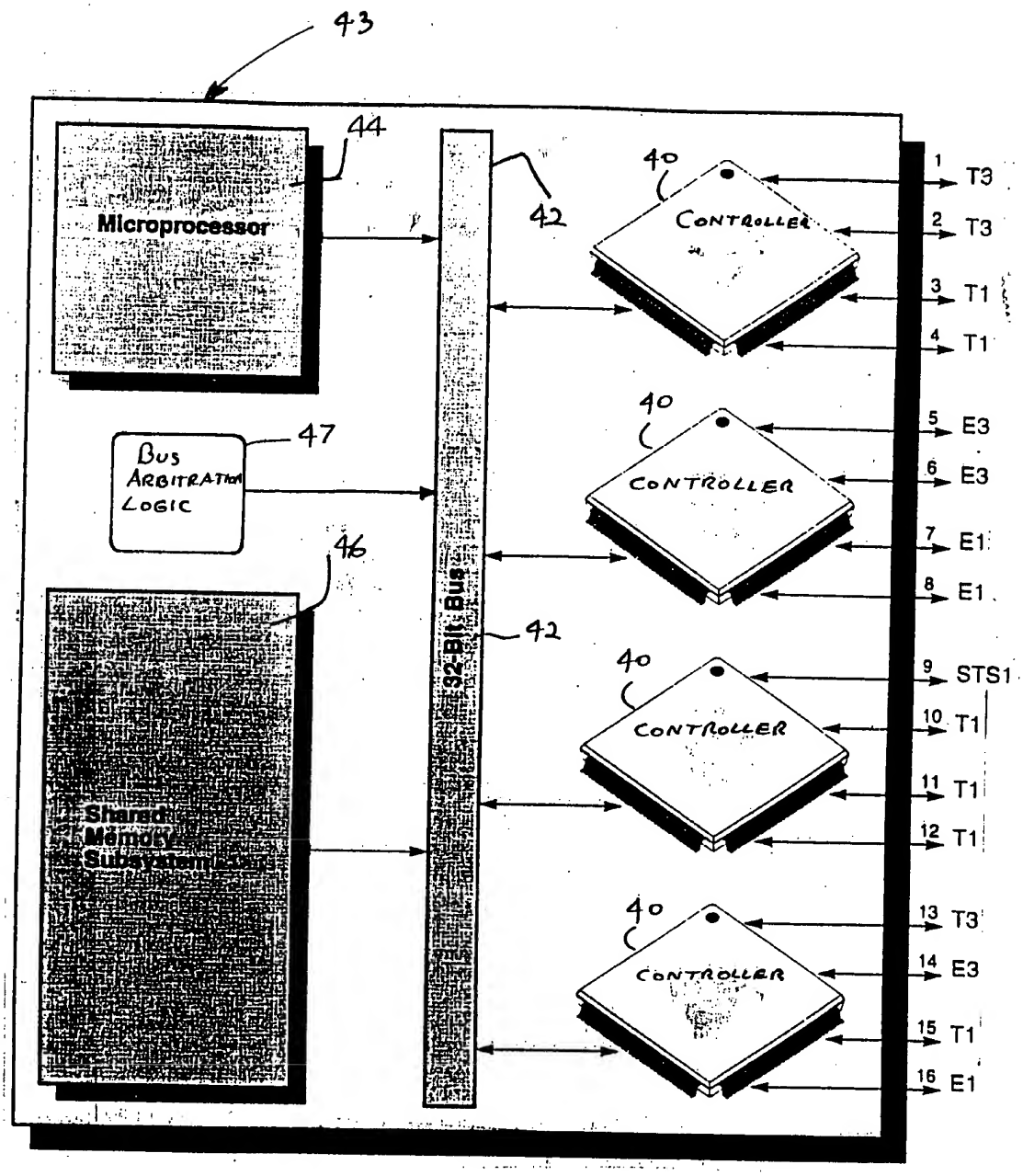

43
44
Microprocessor
42
40
CONTROLLER
1 T3
2 T3
3 T1
4 T1
47
BUS ARBITRATION LOGIC
40
CONTROLLER
5 E3
6 E3
7 E1
8 E1
46
Shared Memory Subsystem
42
32 Bit Bus
40
CONTROLLER
9 STS1
10 T1
11 T1
12 T1
40
CONTROLLER
13 T3
14 E3
15 T1
16 E1

FIG. 1

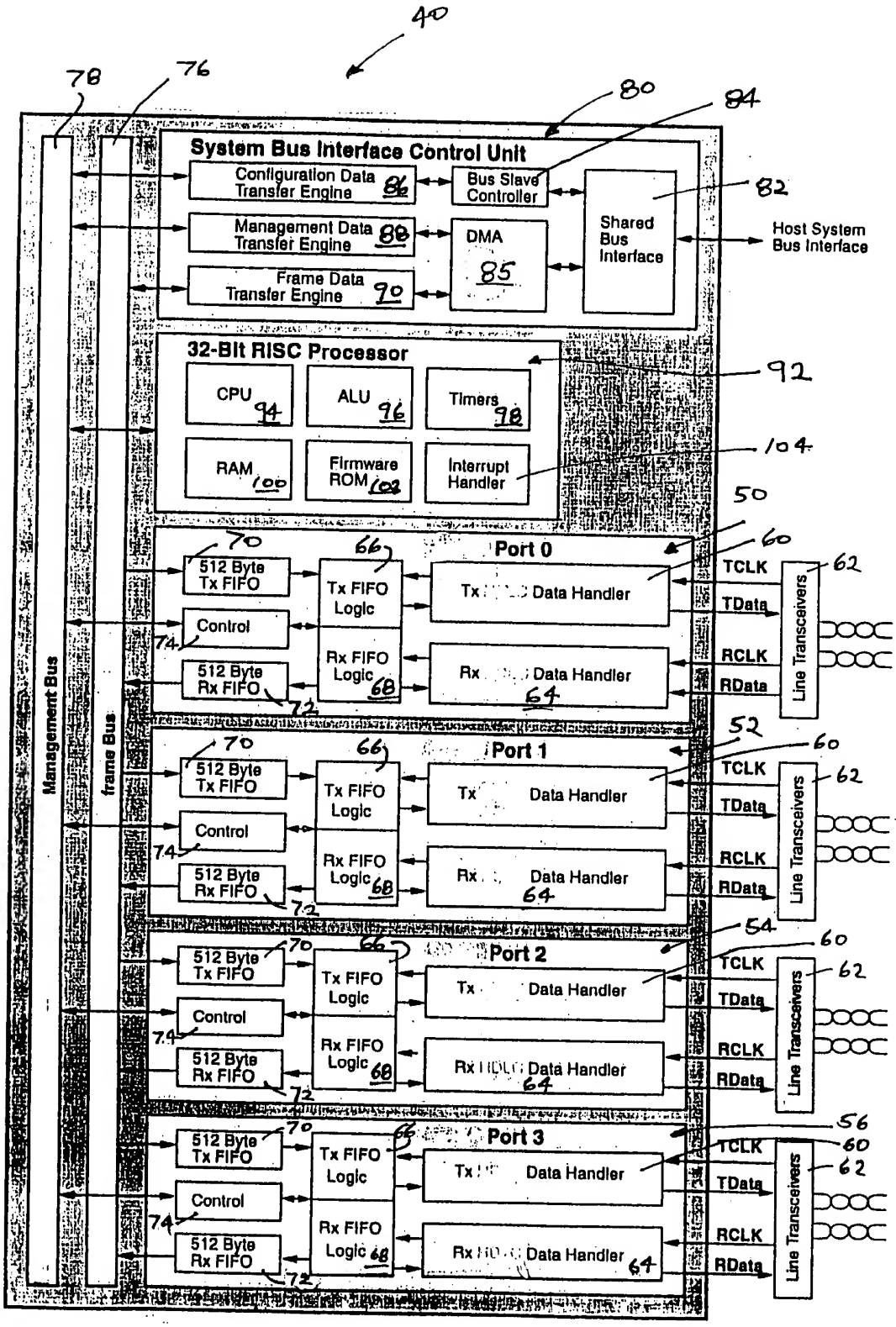

40
78
76
80
84
82
System Bus Interface Control Unit
Configuration Data Transfer Engine
Bus Slave Controller
Management Data Transfer Engine
DMA
Shared Bus Interface
Host System Bus Interface
Frame Data Transfer Engine
85
32-Bit RISC Processor
92
CPU
ALU
Timers
RAM
Firmware ROM
Interrupt Handler
104
50
Port 0
60
62
512 Byte Tx FIFO
Tx FIFO Logic
Tx Data Handler
TCLK
TData
Control
74
Rx FIFO Logic
Rx Data Handler
RCLK
RData
512 Byte Rx FIFO
Line Transceivers
64
52
Port 1
54
Port 2
56
Port 3
Management Bus
frame Bus

FIG. 2

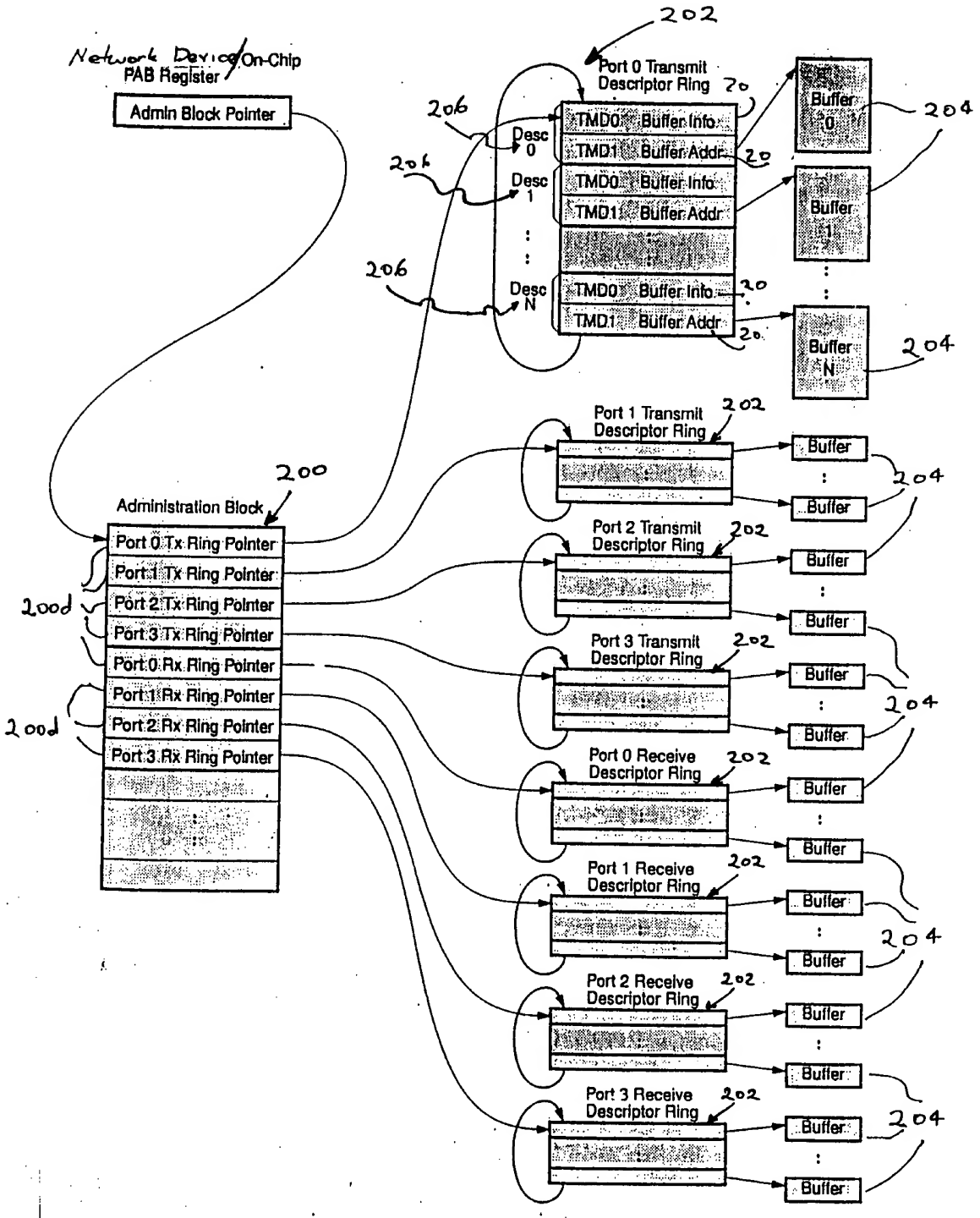

Network Device On-Chip
PAB Register
Admin Block Pointer
Administration Block
200
Port 0 Tx Ring Pointer
Port 1 Tx Ring Pointer
Port 2 Tx Ring Pointer
Port 3 Tx Ring Pointer
Port 0 Rx Ring Pointer
Port 1 Rx Ring Pointer
Port 2 Rx Ring Pointer
Port 3 Rx Ring Pointer
200d
200d
202
Port 0 Transmit Descriptor Ring
206
Desc 0
Desc 1
Desc N
TMD0 Buffer Info
TMD1 Buffer Addr
20
Buffer 0
Buffer 1
Buffer N
204
Port 1 Transmit Descriptor Ring
Port 2 Transmit Descriptor Ring
Port 3 Transmit Descriptor Ring
Port 0 Receive Descriptor Ring
Port 1 Receive Descriptor Ring
Port 2 Receive Descriptor Ring
Port 3 Receive Descriptor Ring
Buffer

FIG. 3

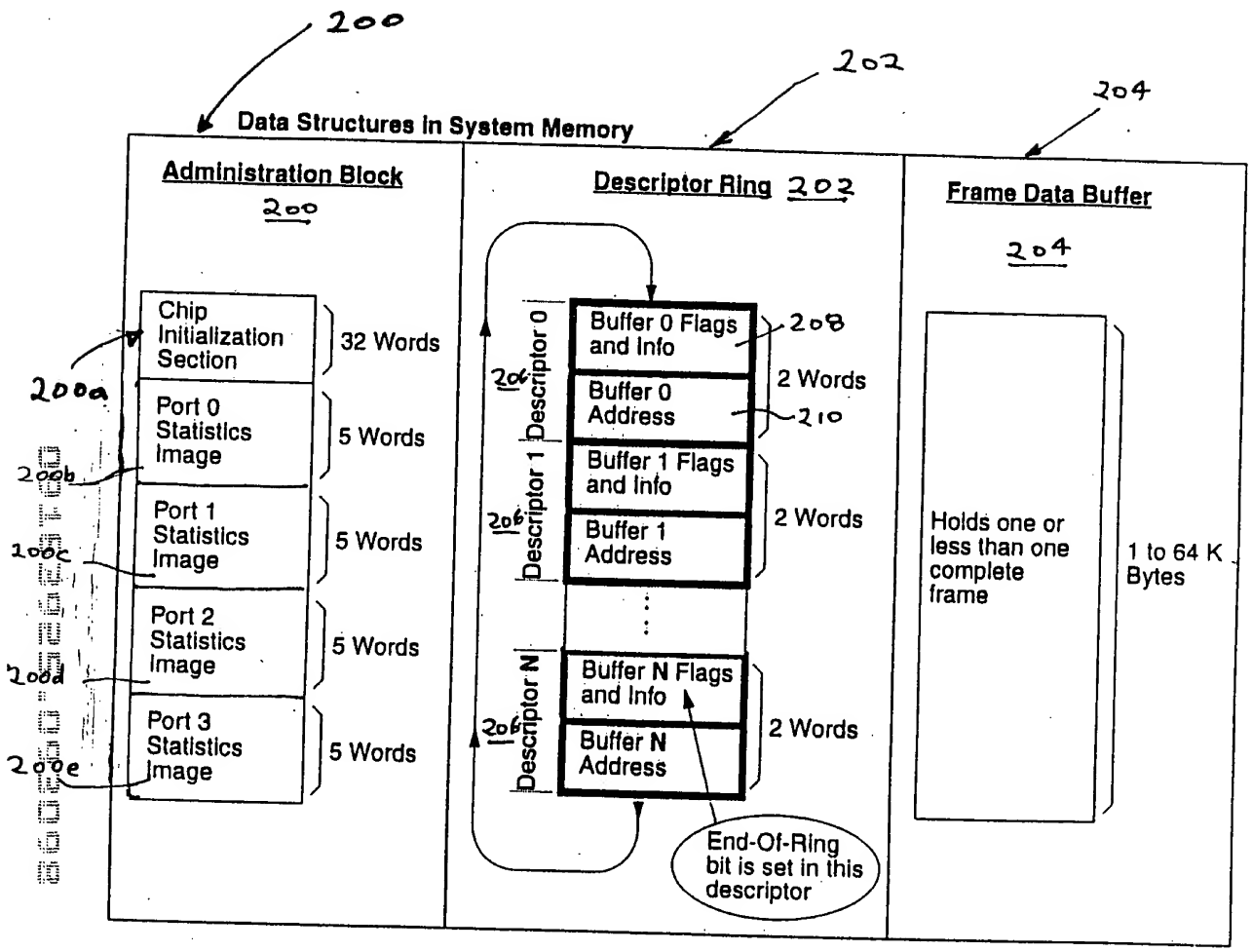

200
202
204
Data Structures in System Memory
Administration Block
200
Descriptor Ring 202
Frame Data Buffer
204
Chip Initialization Section
32 Words
200a
Port 0 Statistics Image
5 Words
200b
Port 1 Statistics Image
5 Words
200c
Port 2 Statistics Image
5 Words
200d
Port 3 Statistics Image
5 Words
200e
Descriptor 0
206
Buffer 0 Flags and Info
208
Buffer 0 Address
210
2 Words
Descriptor 1
206
Buffer 1 Flags and Info
Buffer 1 Address
2 Words
Descriptor N
206
Buffer N Flags and Info
Buffer N Address
2 Words
End-Of-Ring bit is set in this descriptor
Holds one or less than one complete frame
1 to 64 K Bytes

FIG. 4

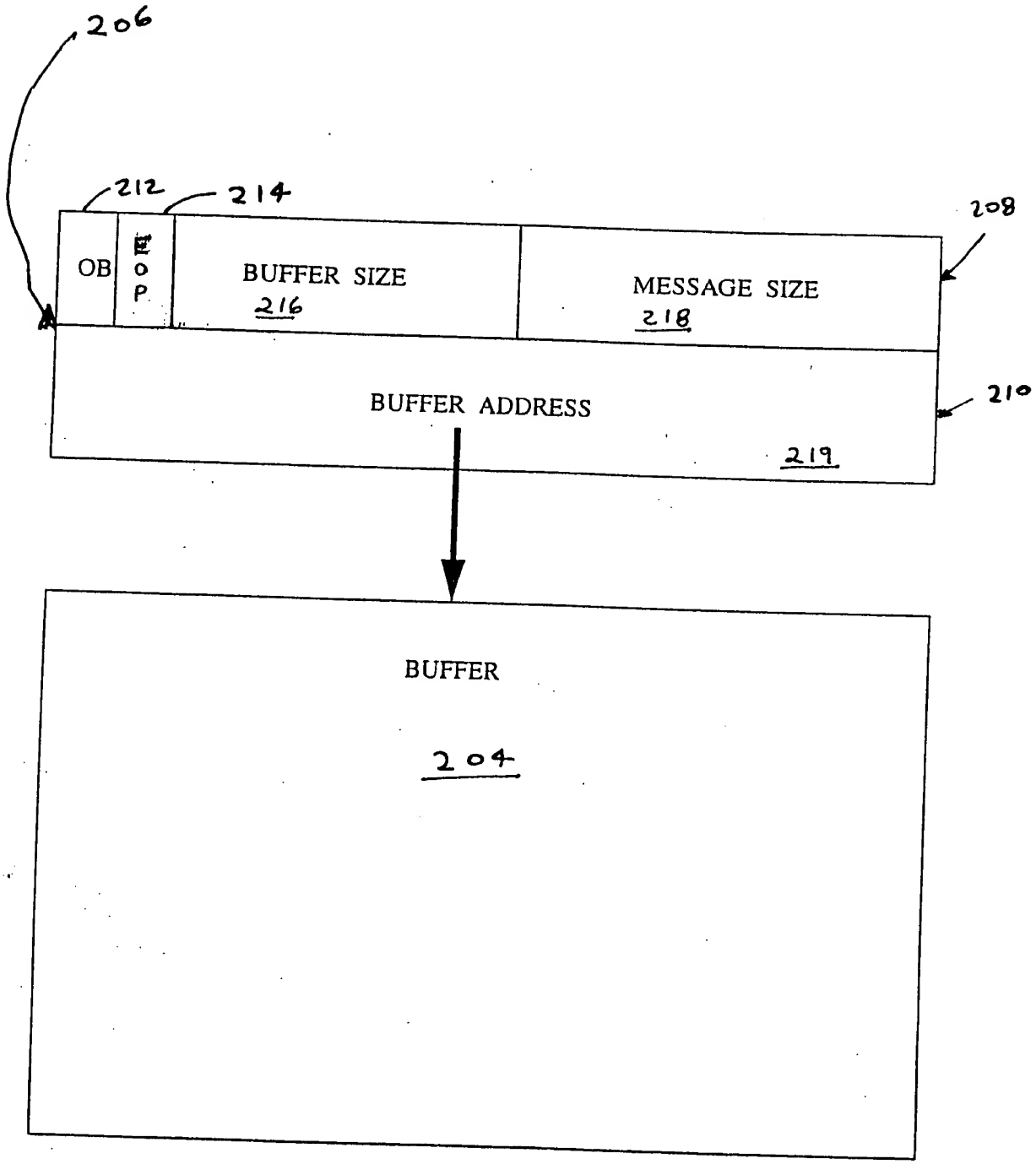

206
212
214
208
OB
EOP
BUFFER SIZE
216
MESSAGE SIZE
218
210
BUFFER ADDRESS
219
BUFFER
204

FIG. 5

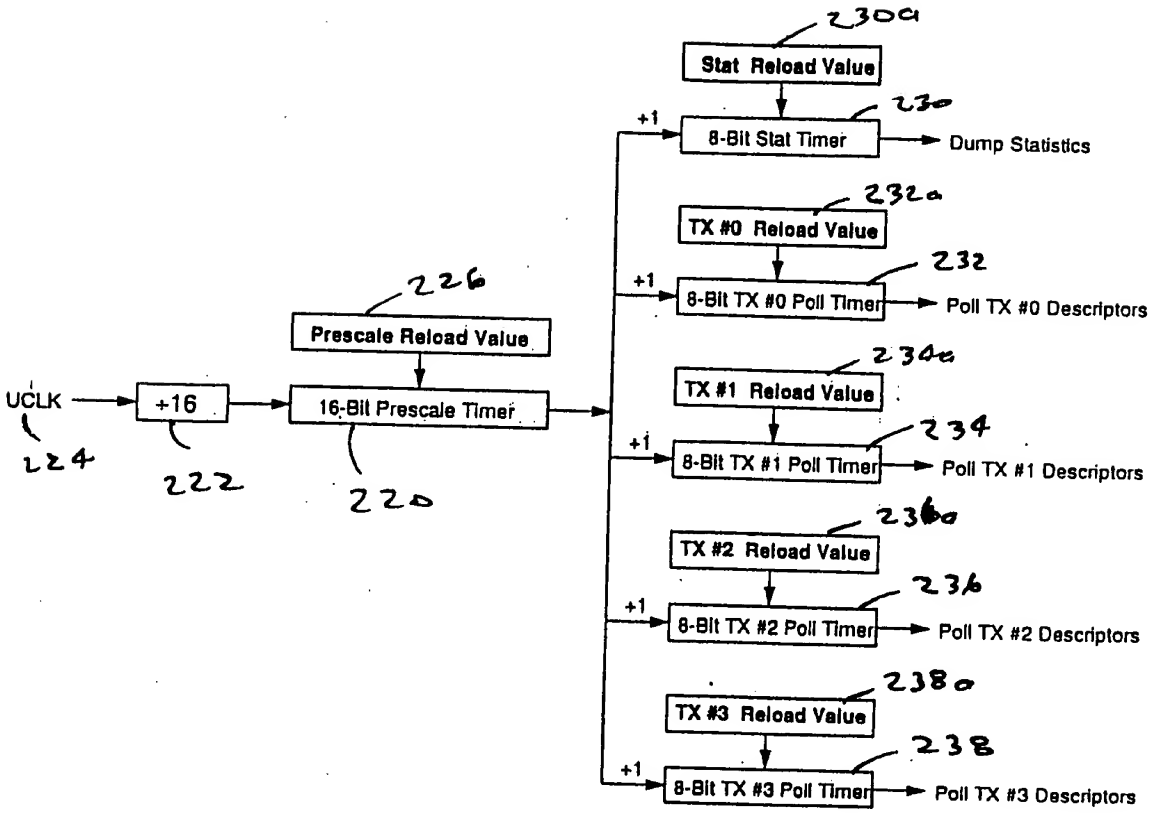

UCLK
124
+16
222
16-Bit Prescale Timer
220
Prescale Reload Value
226
Stat Reload Value
230a
8-Bit Stat Timer
230
Dump Statistics
TX #0 Reload Value
232a
8-Bit TX #0 Poll Timer
232
Poll TX #0 Descriptors
TX #1 Reload Value
234a
8-Bit TX #1 Poll Timer
234
Poll TX #1 Descriptors
TX #2 Reload Value
236a
8-Bit TX #2 Poll Timer
236
Poll TX #2 Descriptors
TX #3 Reload Value
238a
8-Bit TX #3 Poll Timer
238
Poll TX #3 Descriptors
+1

FIG. 6

Relative Address	Byte 3	Byte 2	Byte 1	Byte 0
PAB + 0	Port 0 Tx Top-Of-Ring Descriptor Pointer			
PAB + 4	Port 1 Tx Top-Of-Ring Descriptor Pointer			
PAB + 8	Port 2 Tx Top-Of-Ring Descriptor Pointer			
PAB + 12	Port 3 Tx Top-Of-Ring Descriptor Pointer			
PAB + 16	Port 0 Rx Top-Of-Ring Descriptor Pointer			
PAB + 20	Port 1 Rx Top-Of-Ring Descriptor Pointer			
PAB + 24	Port 2 Rx Top-Of-Ring Descriptor Pointer			
PAB + 28	Port 3 Rx Top-Of-Ring Descriptor Pointer			
PAB + 32	Prescale Timer Reload Value		Stat Timer Reload Value	Timer Enables
PAB + 36	Port 3 TX Poll Timer Reload Value	Port 2 TX Poll Timer Reload Value	Port 1 TX Poll Timer Reload Value	Port 0 TX Poll Timer Reload Value
PAB + 40	Port 3 TX Burst Size	Port 2 TX Burst Size	Port 1 TX Burst Size	Port 0 TX Burst Size
PAB + 44	Port 3 RX Burst Size	Port 2 RX Burst Size	Port 1 RX Burst Size	Port 0 RX Burst Size
PAB + 48	*Reserved*	*Reserved*	UCLK Period (nanoseconds)	Statistics Burst Size
PAB + 52	Port 1 N1		Port 0 N1	
PAB + 56	Port 3 N1		Port 2 N1	
PAB + 60	Port #0 Buffer Size		TX Ring Size	RX Ring Size
PAB + 64	Port #1 Buffer Size		TX Ring Size	RX Ring Size
PAB + 68	Port #2 Buffer Size		TX Ring Size	RX Ring Size
PAB + 72	Port #3 Buffer Size		TX Ring Size	RX Ring Size
PAB + 76	*Reserved*			
PAB + 80	*Reserved*			
PAB + 84	*Reserved*			
PAB + 88	*Reserved*			
PAB + 92	*Reserved*			
PAB + 96	*Reserved*			
PAB + 100	*Reserved*			
PAB + 104	*Reserved*			
PAB + 108	*Reserved*			
PAB + 112	*Reserved*			
PAB + 116	*Reserved*			
PAB + 120	*Reserved*			
PAB + 124	*Reserved*			

200a (PAB + 0 to PAB + 56)

200b (PAB + 60 to PAB + 72)

200c (PAB + 76 to PAB + 124)

200d (PAB + 0 to PAB + 28)

200e (PAB + 32 to PAB + 36)

200f (PAB + 40 to PAB + 44)

200g

200h

200i

FIG 7

Port 0 Relative Address	Byte 3	Byte 2	Byte 1	Byte 0	PORT
PAB + 128	Bad Frames Received				PORT #0
PAB + 132	Aborted Frames				
PAB + 136	Frames Exceeding N1 Received				
PAB + 140	*Reserved*				
PAB + 144	*Reserved*				
PAB + 148	Bad Frames Received				PORT #1
PAB + 152	Aborted Frames				
PAB + 156	Frames Exceeding N1 Received				
PAB + 160	*Reserved*				
PAB + 164	*Reserved*				
PAB + 168	Bad Frames Received				PORT #2
PAB + 172	Aborted Frames				
PAB + 176	Frames Exceeding N1 Received				
PAB + 180	*Reserved*				
PAB + 184	*Reserved*				
PAB + 188	Bad Frames Received				PORT #3
PAB + 192	Aborted Frames				
PAB + 196	Frames Exceeding N1 Received				
PAB + 200	*Reserved*				
PAB + 204	*Reserved*				

FIG. 8

[0x28] PCR -- Primitive Command Register

DMA	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Name	PPA	PPRIM[6:0]							PPARM[7:0]								HPA	HPRIM[6:0]							HPARM[7:0]							
Reset Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Host Access	Read-Clear	Read-Only															Write-Set	Read / Write														
CPC Access	Write-Set	Read / Write															Read-Clear	Read-Only														

Bit #	Field	Name	Description
31	PPA	Provider Primitive Available	**(1=available; 0=no primitive)** Set by the device when the PCR register is written by the firmware. The setting of this bit will also cause the PINT bit of the MIR to be set automatically. This bit is cleared by the DMA when the host reads this register.
30:24	PPRIM	Provider Primitive Command	**(7-bit binary value)** This field is an outgoing (Firmware to Host) primitive command. The meaning is strictly determined by the firmware.
23:16	PPARM	Provider Primitive Parameter	**(8-bit binary value)** This is a firmware defined parameter field corresponding to the provider primitive command.
15	HPA	Host Primitive Available	**(1=available; 0=no primitive)** Set by the device when the PCR register is written by the host. The setting of this bit can result in a CPC interrupt if enabled. This bit is cleared by the the DMA when the firmware reads this register.
14:8	HPRIM	Host Primitive Command	**(7-bit binary value)** This field is an incoming (Host to Firmware) primitive command. The meaning is strictly determined by the firmware.
7:0	HPARM	Host Primitive Parameter	**(8-bit binary value)** This is a firmware defined parameter field corresponding to the host primitive command.

FIG. 8A

[0x2A] MIR – Master Interrupt Register

DMA	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Name	PINT	SPURINT	SDRIFT3	ECN3	FAN3	SHL3	TINT3	RINT3	MERR	PPLOST	SDRIFT2	ECN2	FAN2	SHL2	TINT2	RINT2	SERR	HPLOST	SDRIFT1	ECN1	FAN1	SHL1	TINT1	RINT1	WERR	Spare	SDRIFT0	ECN0	FAN0	SHL0	TINT0	RINT0
Reset Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Host Access	* Read-Clear (No Write)																															
CPC Access	Read-Zeros	Read-Zeros / Write-Ones							Read-Zeros	Read-Zeros / Write-Ones							Read-Zeros	Read-Zeros / Write-Ones							Read-Zeros	Read-Zeros / Write-Ones						

Bit #	Field	Name	Description
31	PINT	Primitive Interrupt	(1=event; 0=no event) Set by the device when the firmware writes a new provider primitive into the Primitive Command Register (upper half).
23	MERR	Memory Error	(1=event; 0=no event) Set by the device when an RTIME Ready Timeout has occurred as defined and established in the System Mode Register (SMR).
15	SERR	System Error	(1=event; 0=no event) Set by the device when an STIME System Timeout has occurred as defined and established in the System Mode Register (SMR).
7	WERR	Configuration Write Error	(1=event; 0=no event) Set by the device when the host has attempted to write to a register location which is inaccessible by the host. This bit will never be set when host access is unlocked via the KEY field of the LOCK Register.
30	SPURINT	Spurious CPC Interrupt	(1=event; 0=no event) Set by the CPC firmware indicating the reception of an invalid internal CPC interrupt. This is a device hardware fault and should never occur.
22	PPLOST	Provider Primitive Lost	(1=event; 0=no event) Set by the device when the firmware writes a new provider primitive over one that has not yet been read by the host. This condition is detected by testing the PPA bit of the Primitive Command Register (PCR).
14	HPLOST	Host Primitive Lost	(1=event; 0=no event) Set by the device when the host writes a new host primitive over one that has not yet been read by the firmware. This condition is detected by testing the HPA bit of the Primitive Command Register (PCR).
6	Spare		
29, 21, 13, 5	SDRIFT	Port *n* Statistic Drift	(1=event; 0=no event) Set by the CPC firmware when conditions in the corresponding port have been reached where statistical information might be lost. This will only happen when receive congestion is occurring such that frames are being lost due to lack of available space in the port's receive FIFO.
28, 20, 12, 4	ECN	Port *n* Early Congestion Notification	(1=event; 0=no event) Set by the CPC firmware for advanced host notification of congestion in the corresponding port's receiver. Congestion occurs when a unit is forced to drop a received frame due to lack of available space in the Rx FIFO.
27, 19, 11, 3	FAN	Frame Address Notification	(1=event; 0=no event) Set by the device to notify the host that the address fields are present in the frame buffer.
26, 18, 10, 2	SHL	Port *n* Statistic Half-Life	(1=event; 0=no event) Set by the CPC firmware when one or more of the correspond port's statistics has passed the half-full mark – defined as the most-significant-bit of a statistic changing polarity (0-to-1 or 1-to-0) due to the last update.
25, 17, 9, 1	TINT	Port *n* Transmit Interrupt	(1=event; 0=no event) Set by the device whenever the transmission of one or more frames has been completed. For a successfully transmitted frame this interrupt signals that the frame has cleared the chip.
24, 16, 8, 0	RINT	Port *n* Receive Interrupt	(1=event; 0=no event) Set by the device whenever a receive frame has been completely transferred from the corresponding port to the host system. This means a frame has been transferred to System Memory. If the reception of bad frames (RBUFF=0 in the SMR register) has been disabled then no RINT will be generated in the event of bad frames.

FIG. 8B

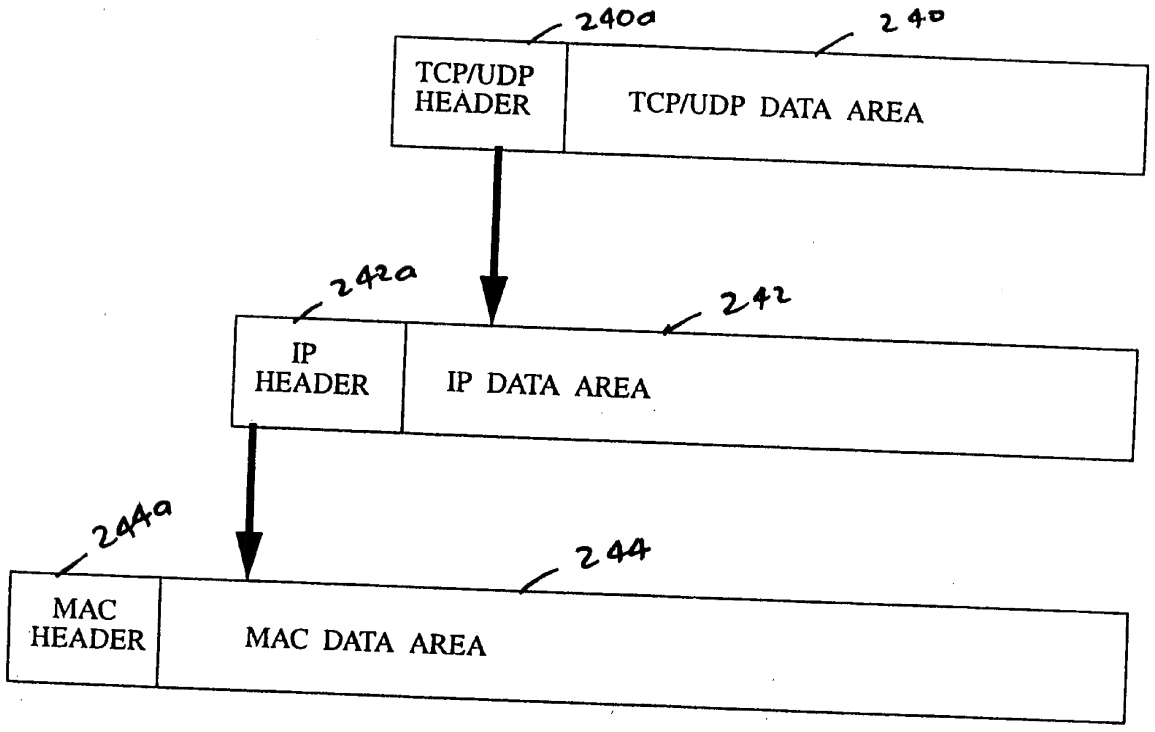

240a
240
TCP/UDP HEADER
TCP/UDP DATA AREA
242a
242
IP HEADER
IP DATA AREA
244a
244
MAC HEADER
MAC DATA AREA

FIG. 9

802.3 DATALINK LAYER HEADER (18 Bytes)

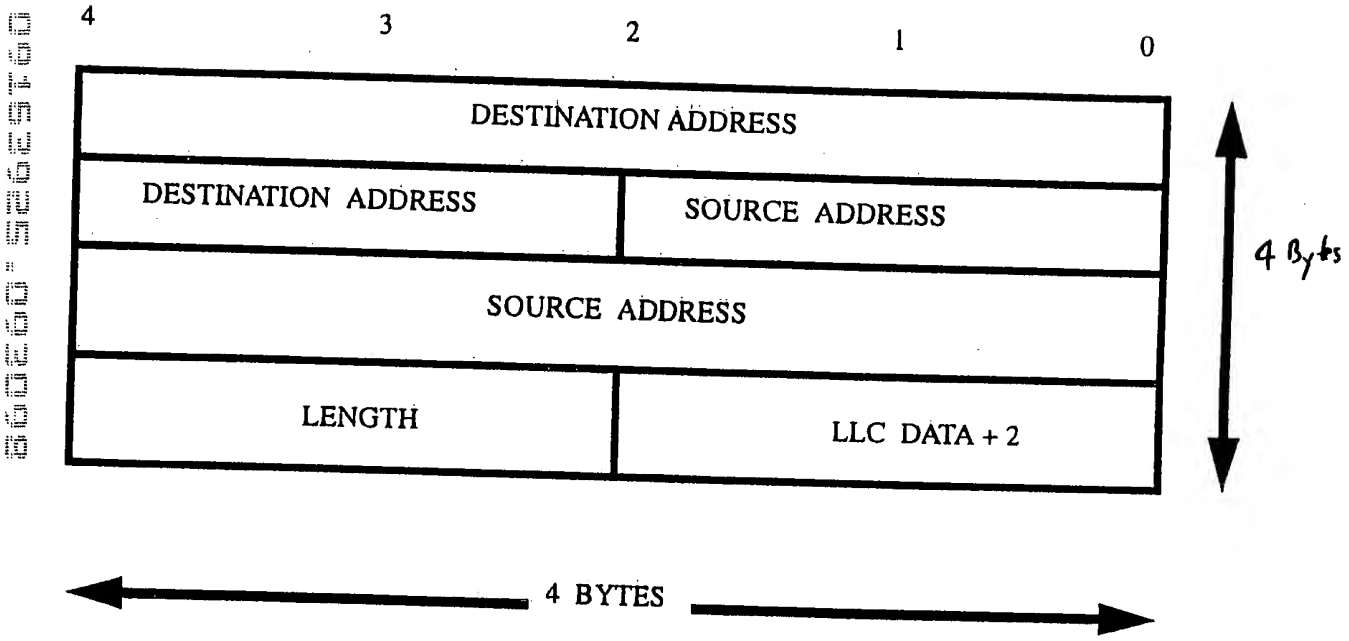


FIG. 10

INTERNET IP HEADER (20 Bytes)

VER/HEADER	TYPE OF SERVICE	16-BIT TOTAL LENGTH (IN BYTES)
16-BIT IDENTIFICATON		3-BIT FLAGS/13-BIT FRAGMENT OFFSET
TTL	8-BIT PROTOCOL	16-BIT HEADER CHECKSUM
32-BIT SOURCE IP ADDRESS		
32-BIT DESTINATION IP ADDRESS		
(OPTIONS - IF ANY)		

FIG. 11

TCP HEADER (20 BYTES)

16-BIT SOURCE PORT	16-BIT DESTINATION PORT
32-BIT SEQUENCE NUMBER	
32-BIT ACKNOWLEDGMENT NUMBER	
URG/ACK/PSH/RST/SYN/FIN	16-BIT WINDOW SIZE
16-BIT TCP CHECKSUM	16-BIT URGENT POINTER

FIG. 12

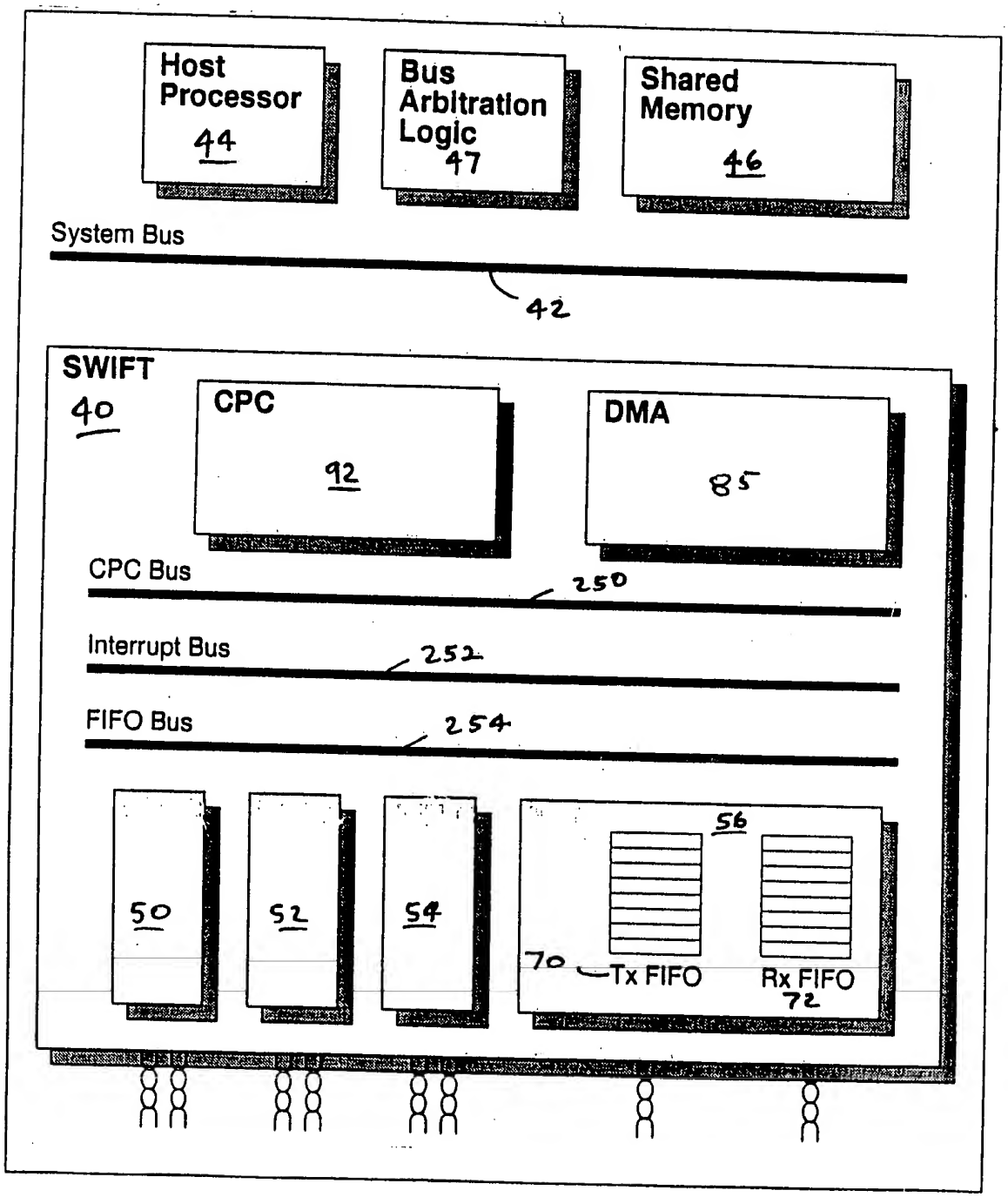

Host Processor
44
Bus Arbitration Logic
47
Shared Memory
46
System Bus
42
SWIFT
40
CPC
92
DMA
85
CPC Bus
250
Interrupt Bus
252
FIFO Bus
254
50
52
54
56
70
Tx FIFO
Rx FIFO
72

FIG. 13

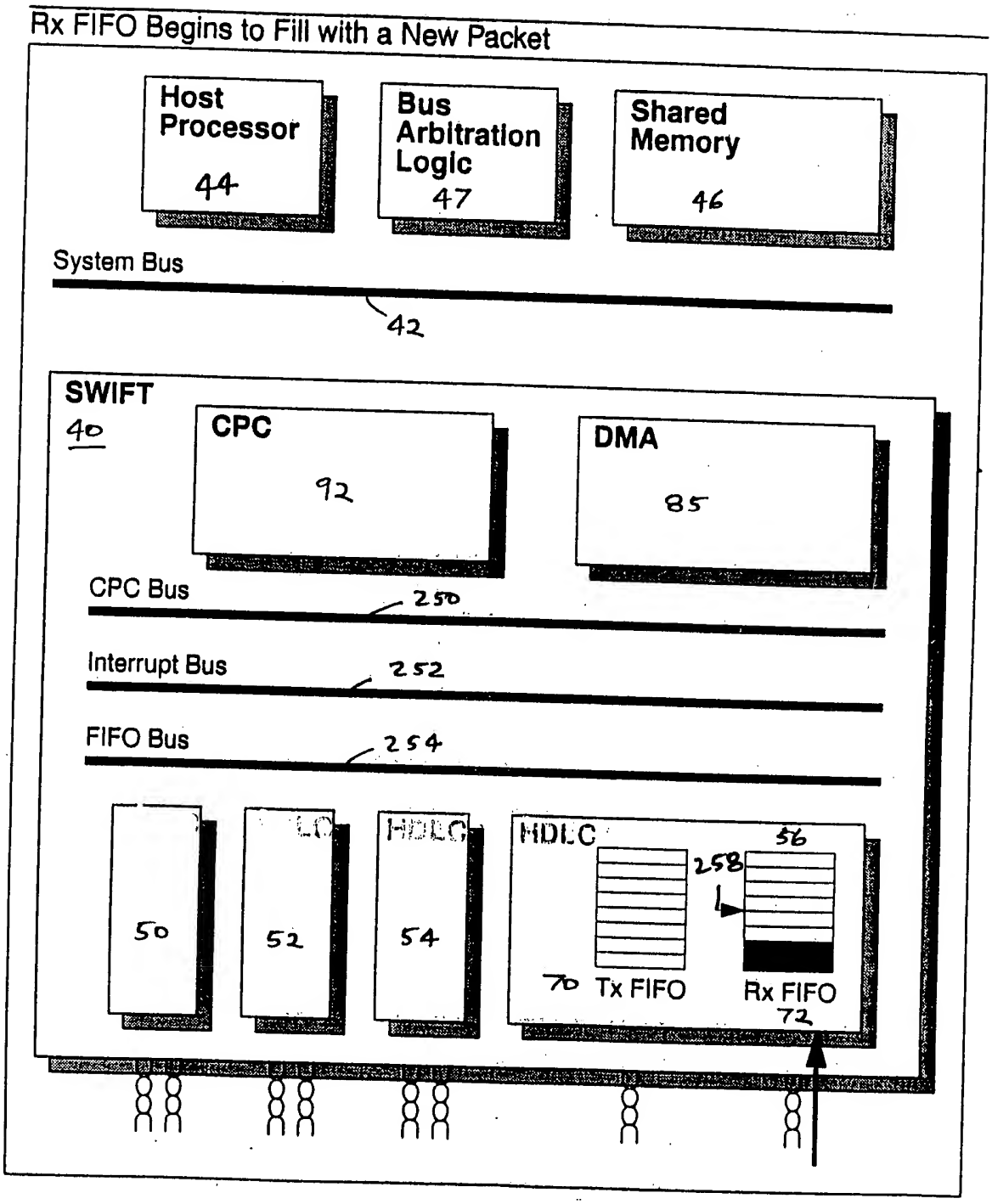

Rx FIFO Begins to Fill with a New Packet
Host Processor
44
Bus Arbitration Logic
47
Shared Memory
46
System Bus
42
SWIFT
40
CPC
92
DMA
85
CPC Bus
250
Interrupt Bus
252
FIFO Bus
254
50
52
54
HDLC
56
258
70
Tx FIFO
Rx FIFO
72

FIG. 14

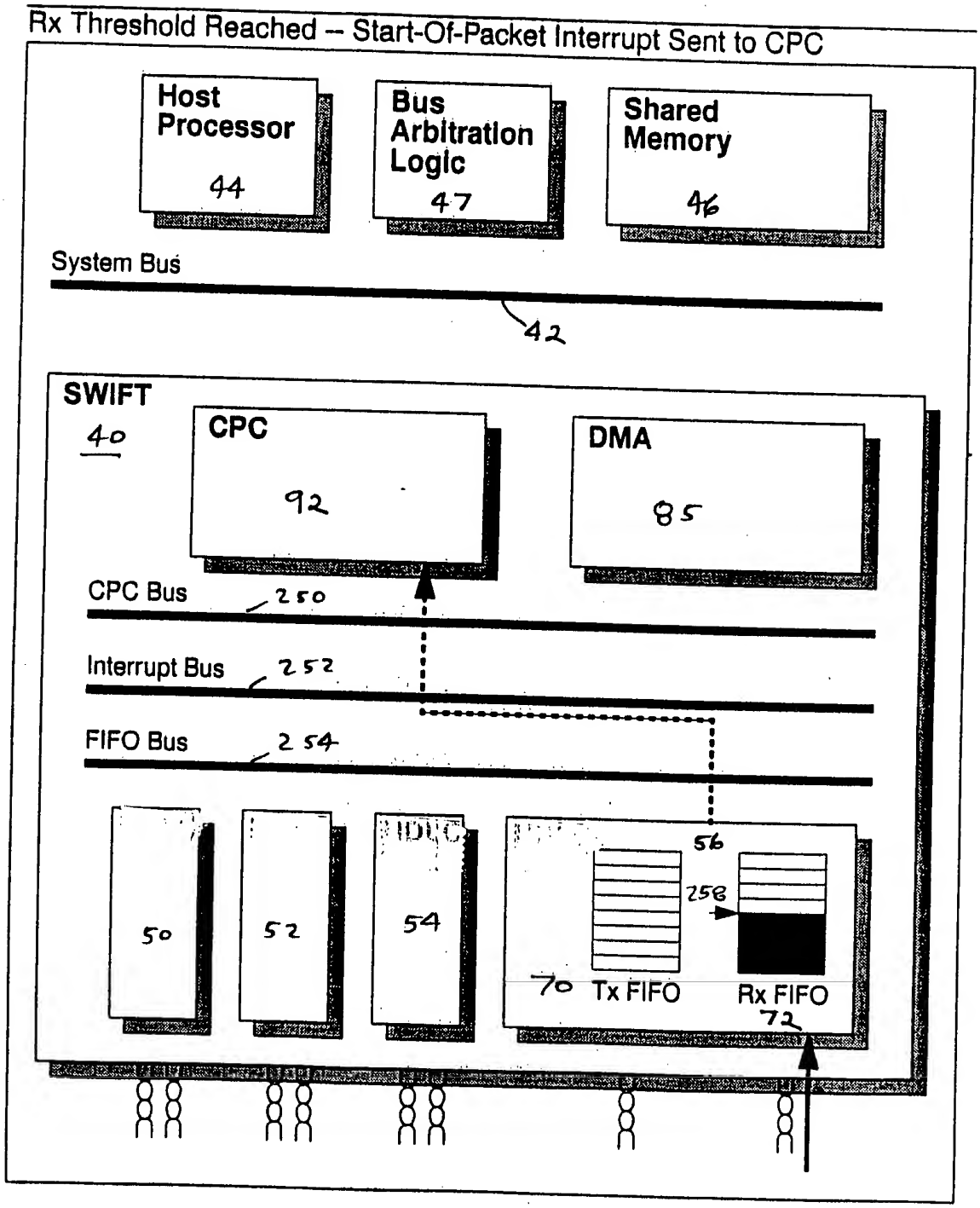

Rx Threshold Reached -- Start-Of-Packet Interrupt Sent to CPC
Host Processor
44
Bus Arbitration Logic
47
Shared Memory
46
System Bus
42
SWIFT
40
CPC
92
DMA
85
CPC Bus
250
Interrupt Bus
252
FIFO Bus
254
50
52
IDLC
54
56
258
70
Tx FIFO
Rx FIFO
72

FIG. 15

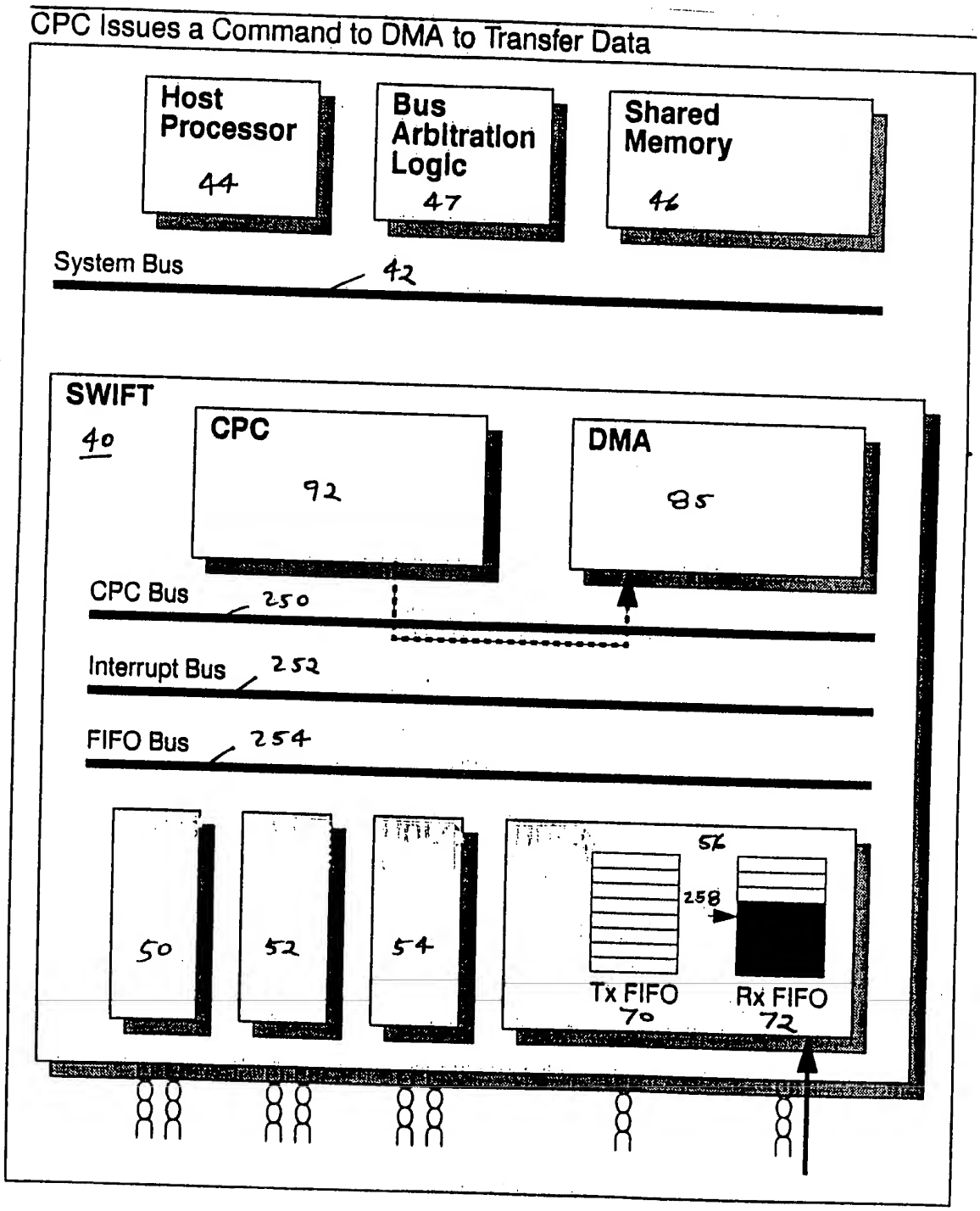

CPC Issues a Command to DMA to Transfer Data
Host Processor
44
Bus Arbitration Logic
47
Shared Memory
46
System Bus
42
SWIFT
40
CPC
92
DMA
85
CPC Bus
250
Interrupt Bus
252
FIFO Bus
254
50
52
54
56
258
Tx FIFO
70
Rx FIFO
72

FIG. 16

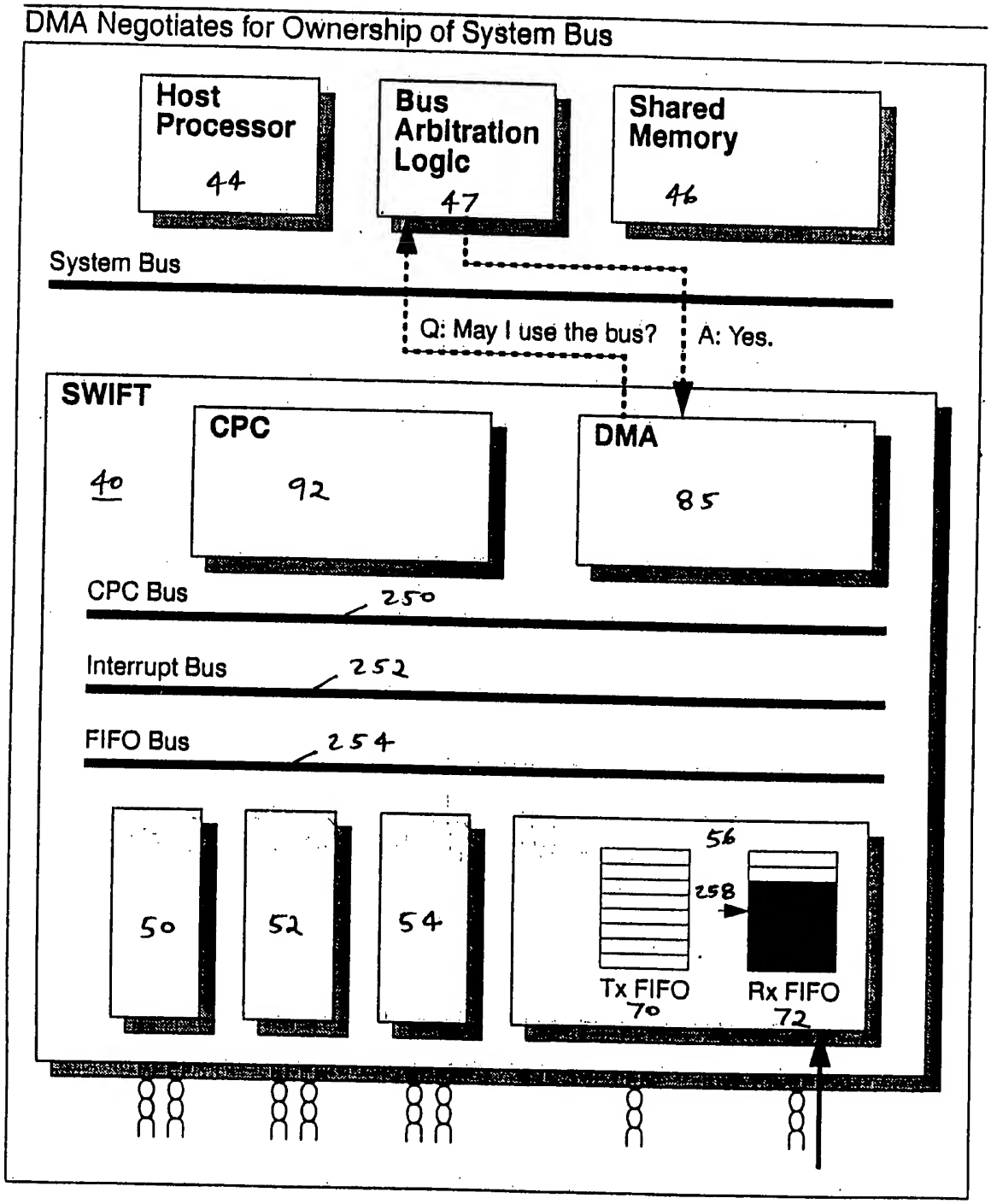

DMA Negotiates for Ownership of System Bus
Host Processor
44
Bus Arbitration Logic
47
Shared Memory
46
System Bus
Q: May I use the bus?
A: Yes.
SWIFT
40
CPC
92
DMA
85
CPC Bus
250
Interrupt Bus
252
FIFO Bus
254
50
52
54
56
258
Tx FIFO
70
Rx FIFO
72

FIG. 17

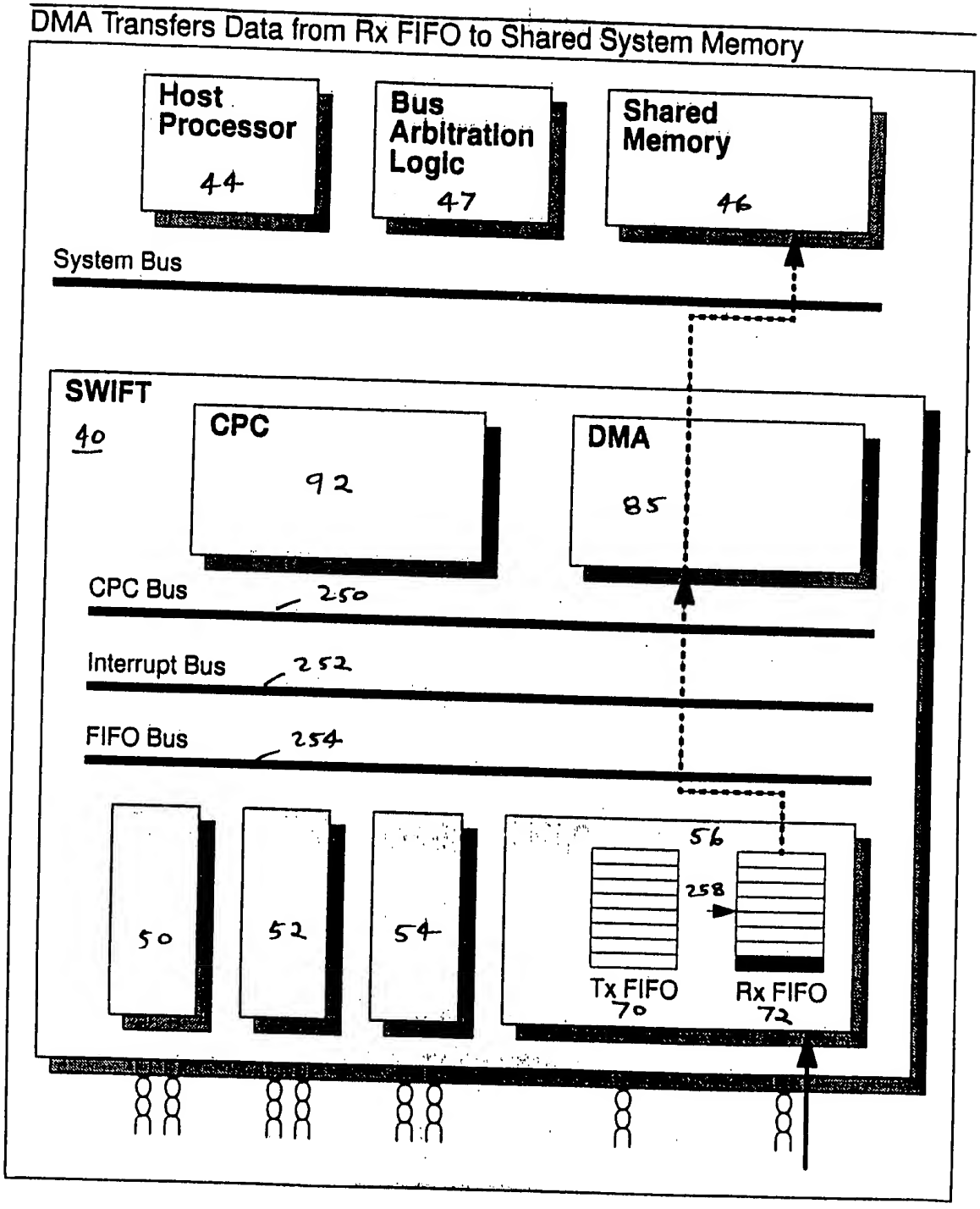

DMA Transfers Data from Rx FIFO to Shared System Memory
Host Processor
44
Bus Arbitration Logic
47
Shared Memory
46
System Bus
SWIFT
40
CPC
92
DMA
85
CPC Bus
250
Interrupt Bus
252
FIFO Bus
254
50
52
54
56
258
Tx FIFO
70
Rx FIFO
72

FIG. 18

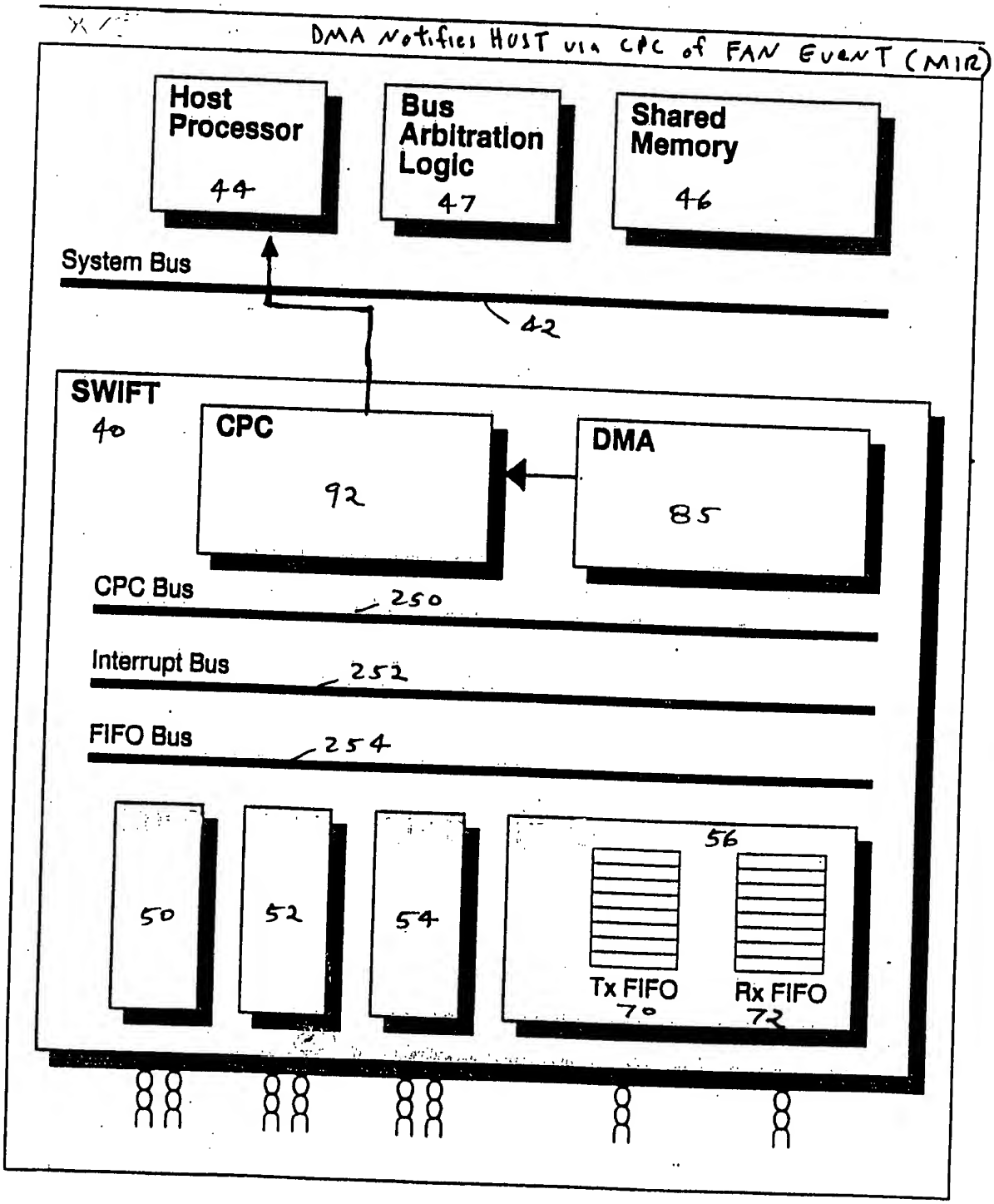

DMA Notifies HOST via CPC of FAN EVENT (MIR)
Host Processor
44
Bus Arbitration Logic
47
Shared Memory
46
System Bus
42
SWIFT
40
CPC
92
DMA
85
CPC Bus
250
Interrupt Bus
252
FIFO Bus
254
50
52
54
56
Tx FIFO
70
Rx FIFO
72

FIG. 19

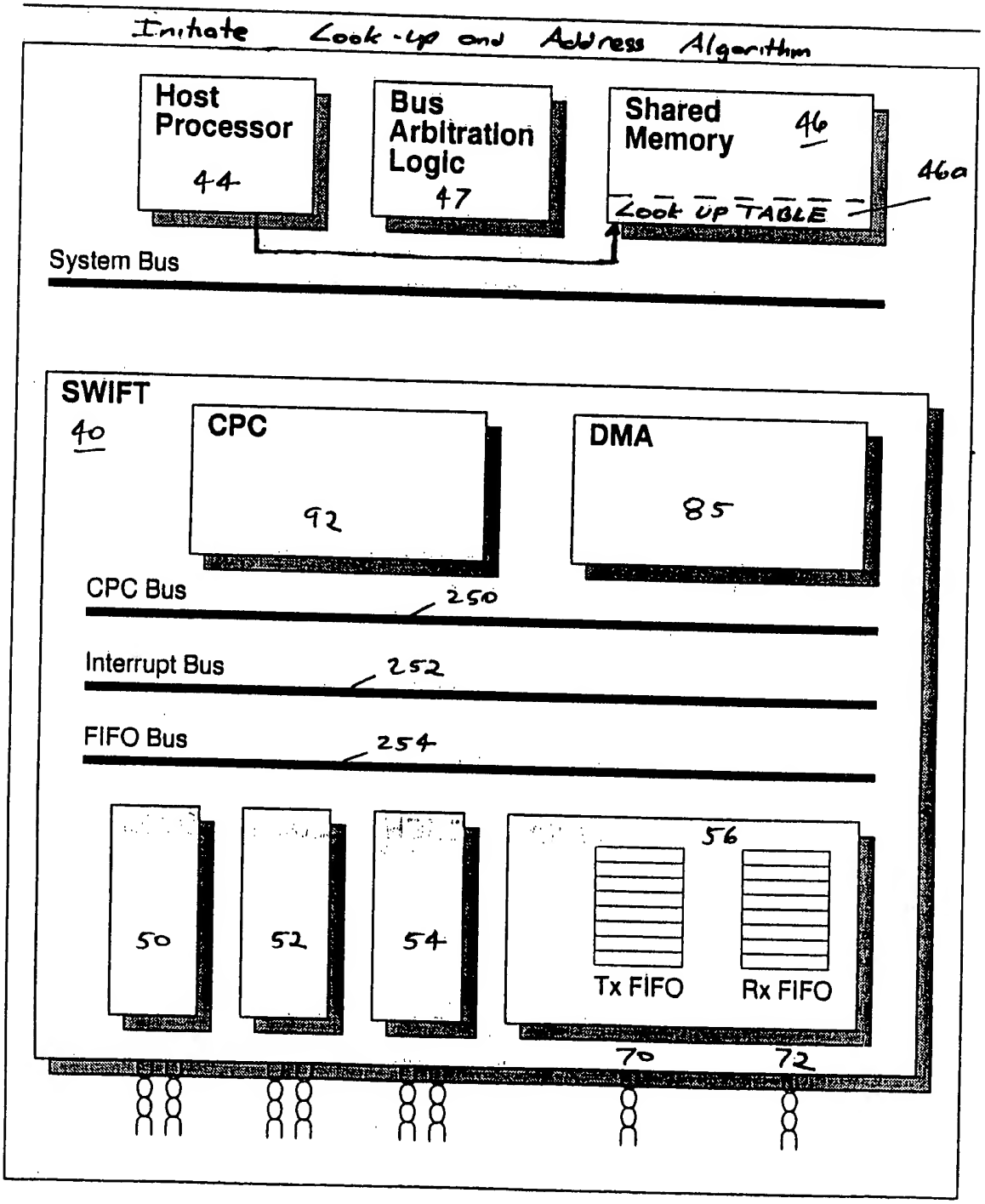

Initiate Look-up and Address Algorithm
Host Processor
44
Bus Arbitration Logic
47
Shared Memory
46
46a
Look UP TABLE
System Bus
SWIFT
40
CPC
92
DMA
85
CPC Bus
250
Interrupt Bus
252
FIFO Bus
254
50
52
54
56
Tx FIFO
Rx FIFO
70
72

FIG. 20

FRAME ADDRESS NOTIFICATION (FAN)

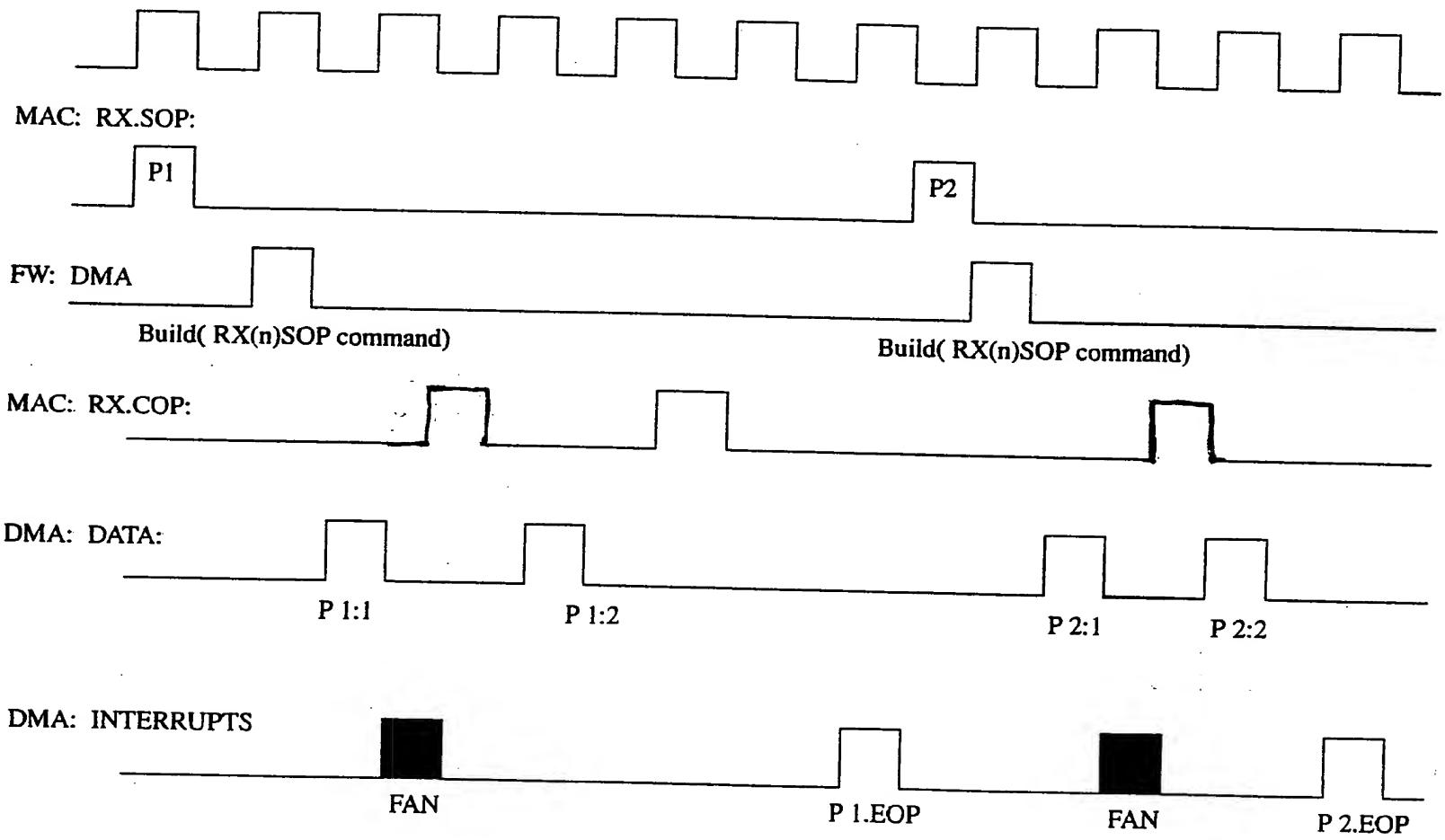


FIG. 21

CONVENTIONAL FIFO FLOW-CONTROL VERSUS LAWM

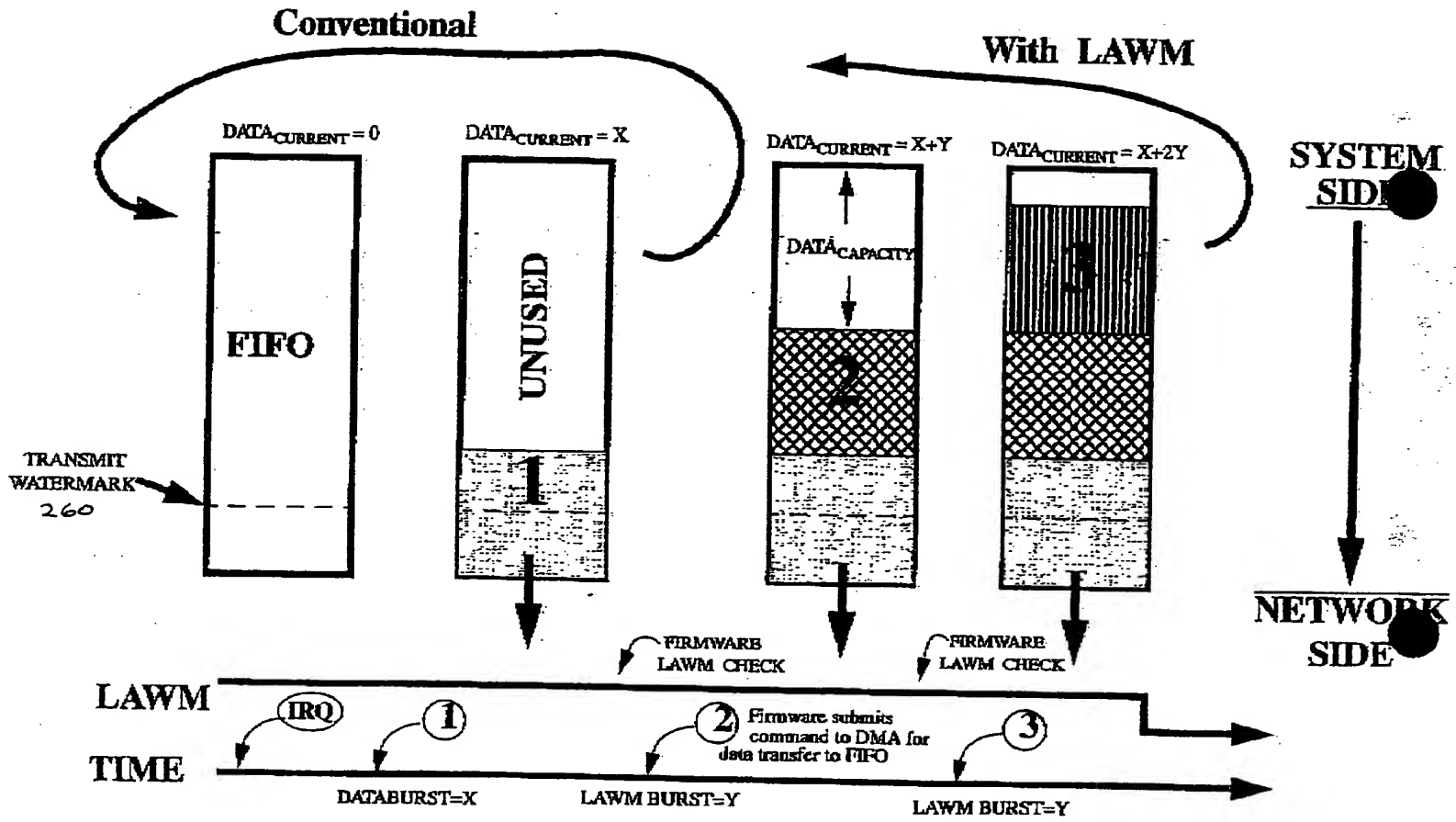
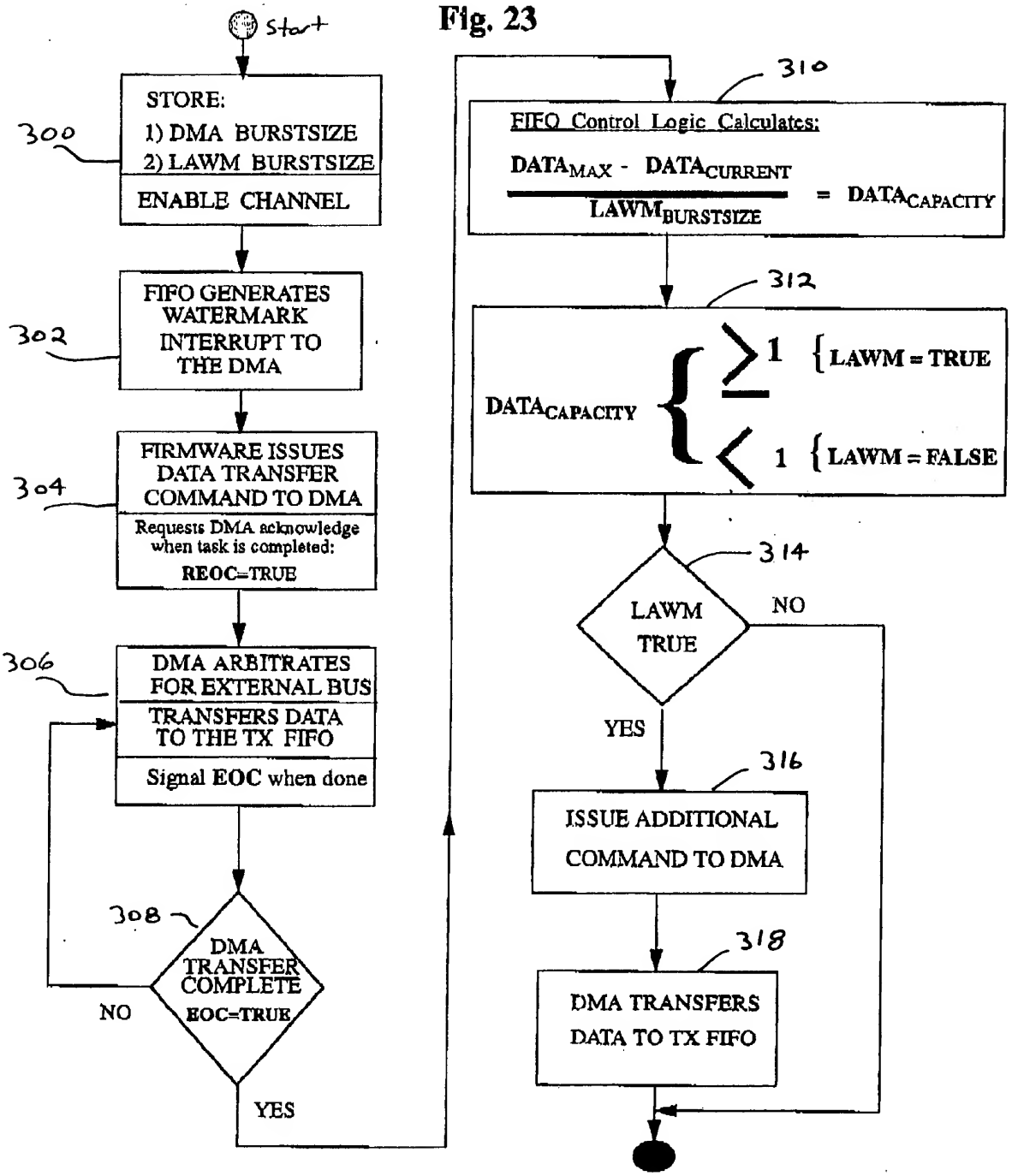


Fig. 22

Fig. 23


Start
STORE:
1) DMA BURSTSIZE
2) LAWM BURSTSIZE
ENABLE CHANNEL
300
FIFO GENERATES WATERMARK INTERRUPT TO THE DMA
302
FIRMWARE ISSUES DATA TRANSFER COMMAND TO DMA
Requests DMA acknowledge when task is completed:
REOC=TRUE
304
DMA ARBITRATES FOR EXTERNAL BUS
TRANSFERS DATA TO THE TX FIFO
Signal EOC when done
306
DMA TRANSFER COMPLETE
EOC=TRUE
308
NO
YES
FIFO Control Logic Calculates:
DATA_MAX - DATA_CURRENT / LAWM_BURSTSIZE = DATA_CAPACITY
310
DATA_CAPACITY ≥ 1 { LAWM = TRUE
DATA_CAPACITY < 1 { LAWM = FALSE
312
LAWM TRUE
314
NO
YES
ISSUE ADDITIONAL COMMAND TO DMA
316
DMA TRANSFERS DATA TO TX FIFO
318

INTERRUPT-MEDIATED FRAME TRANSMISSION

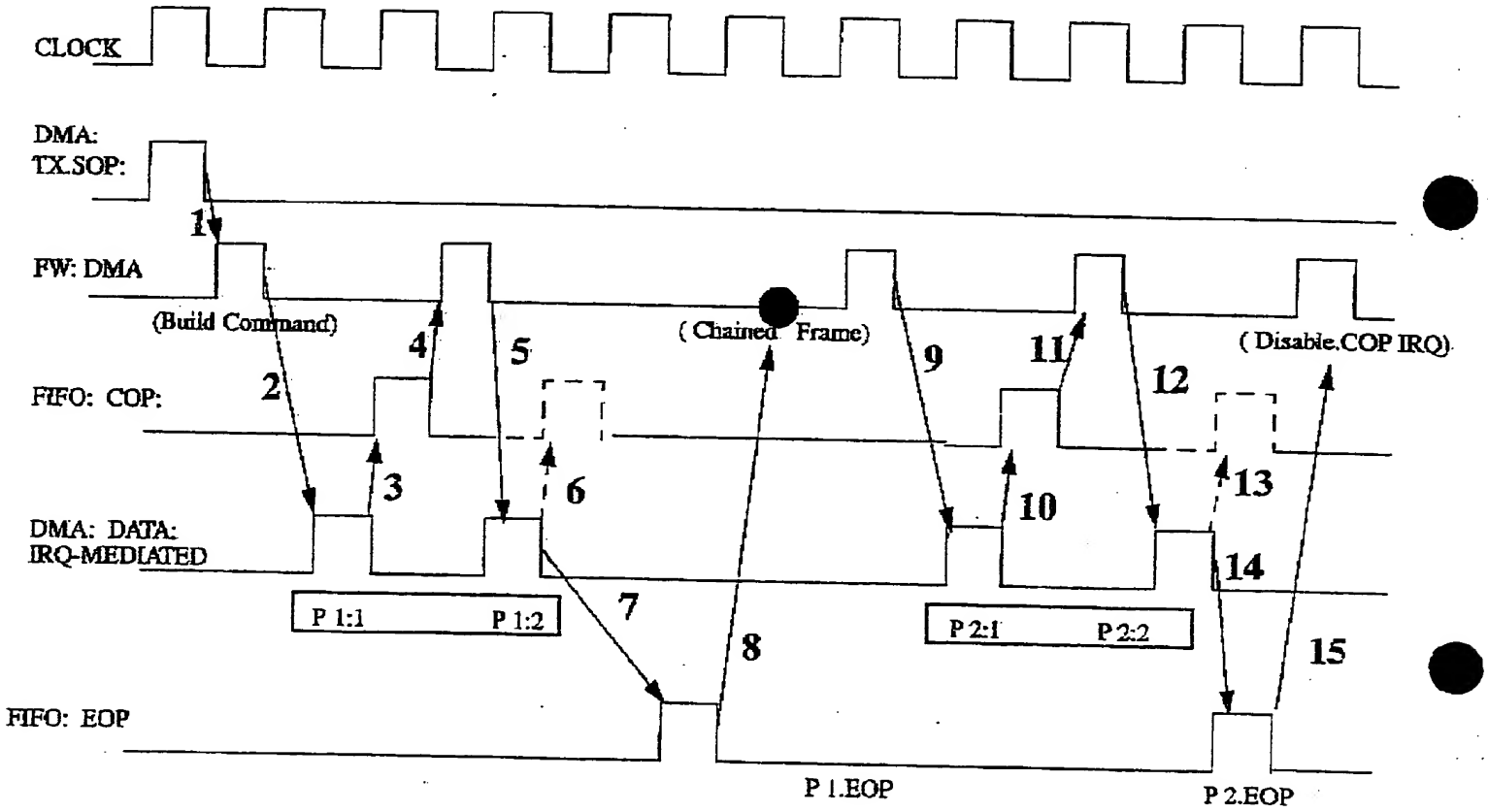


FIG. 24a

LAWM-MEDIATED FRAME TRANSMISSION

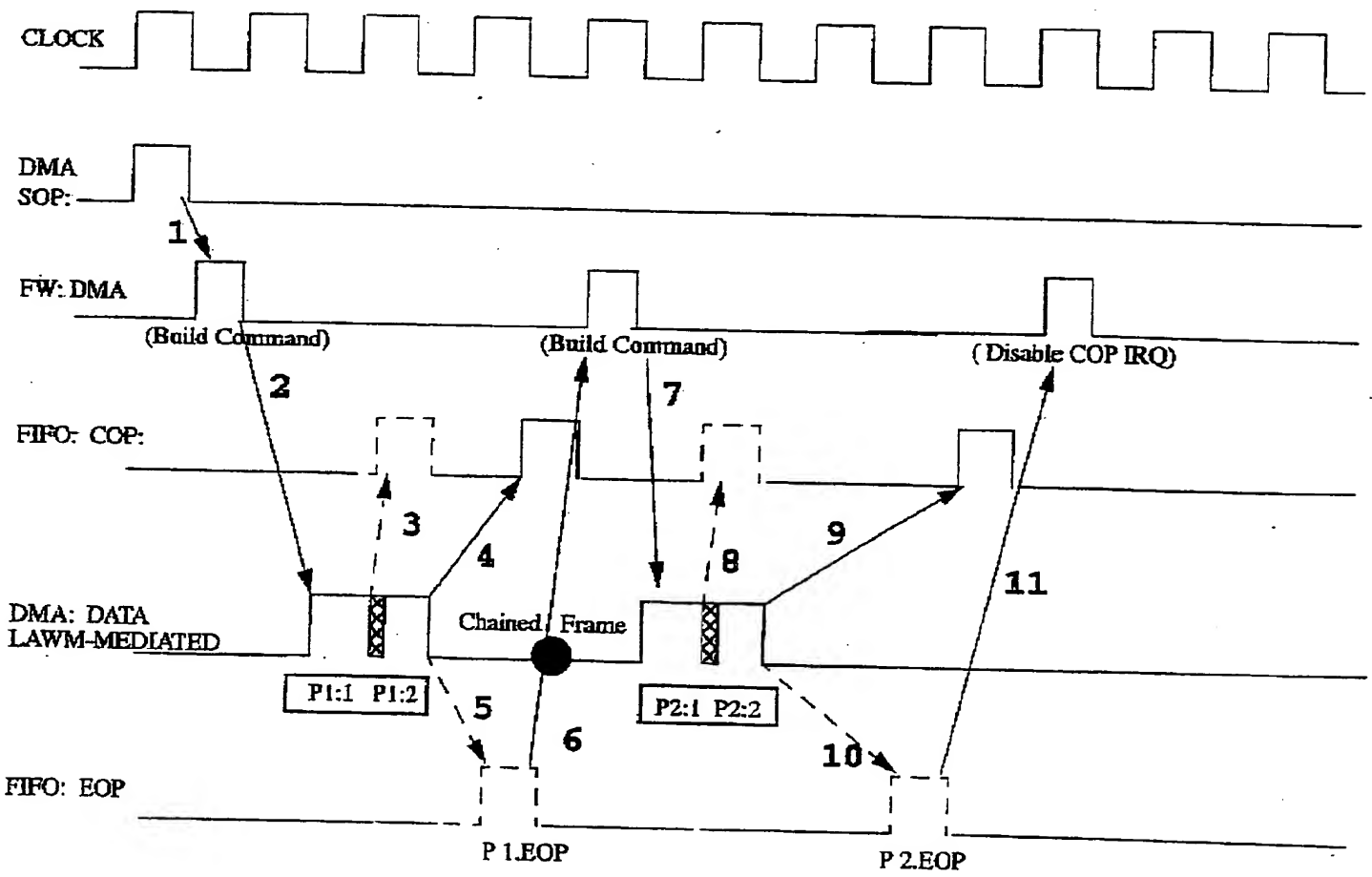
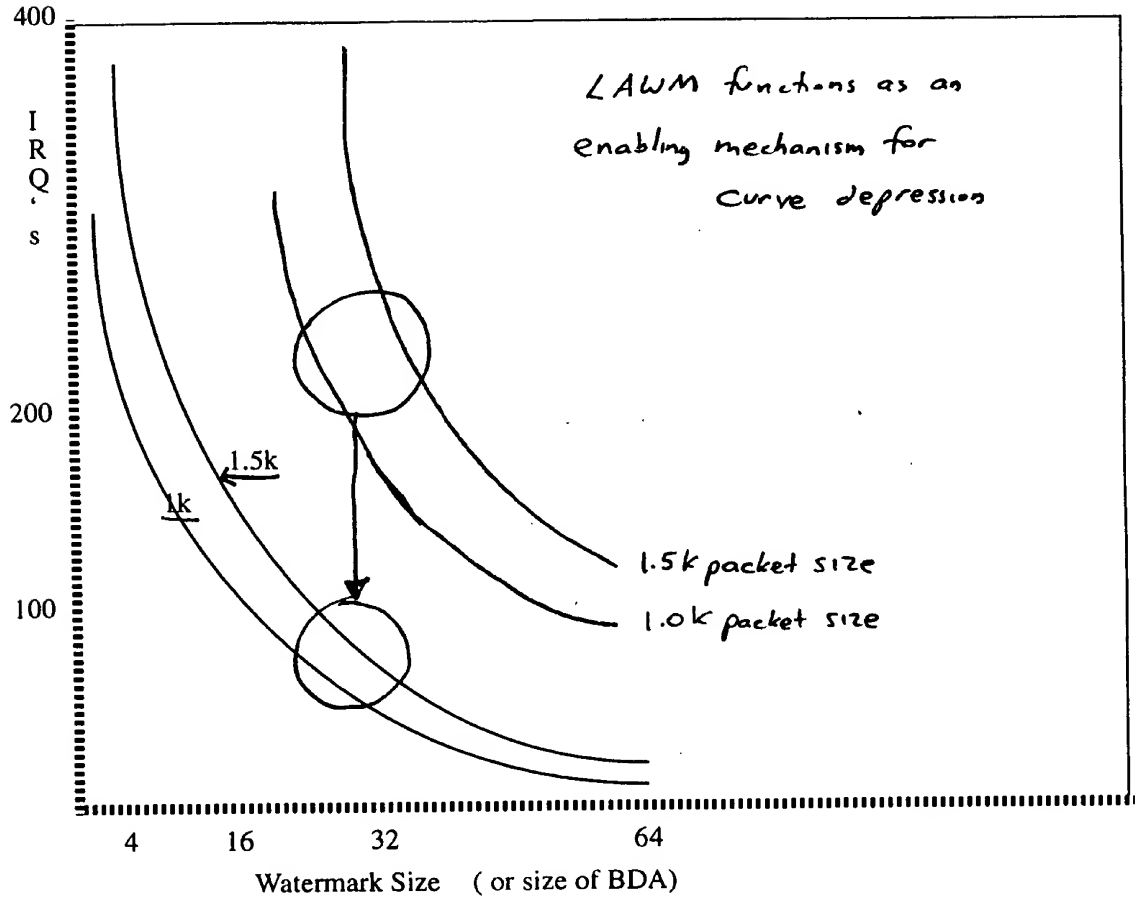


FIG. 24b

FIG. 25

Watermark effects on IRQ generation with regard to packet size:



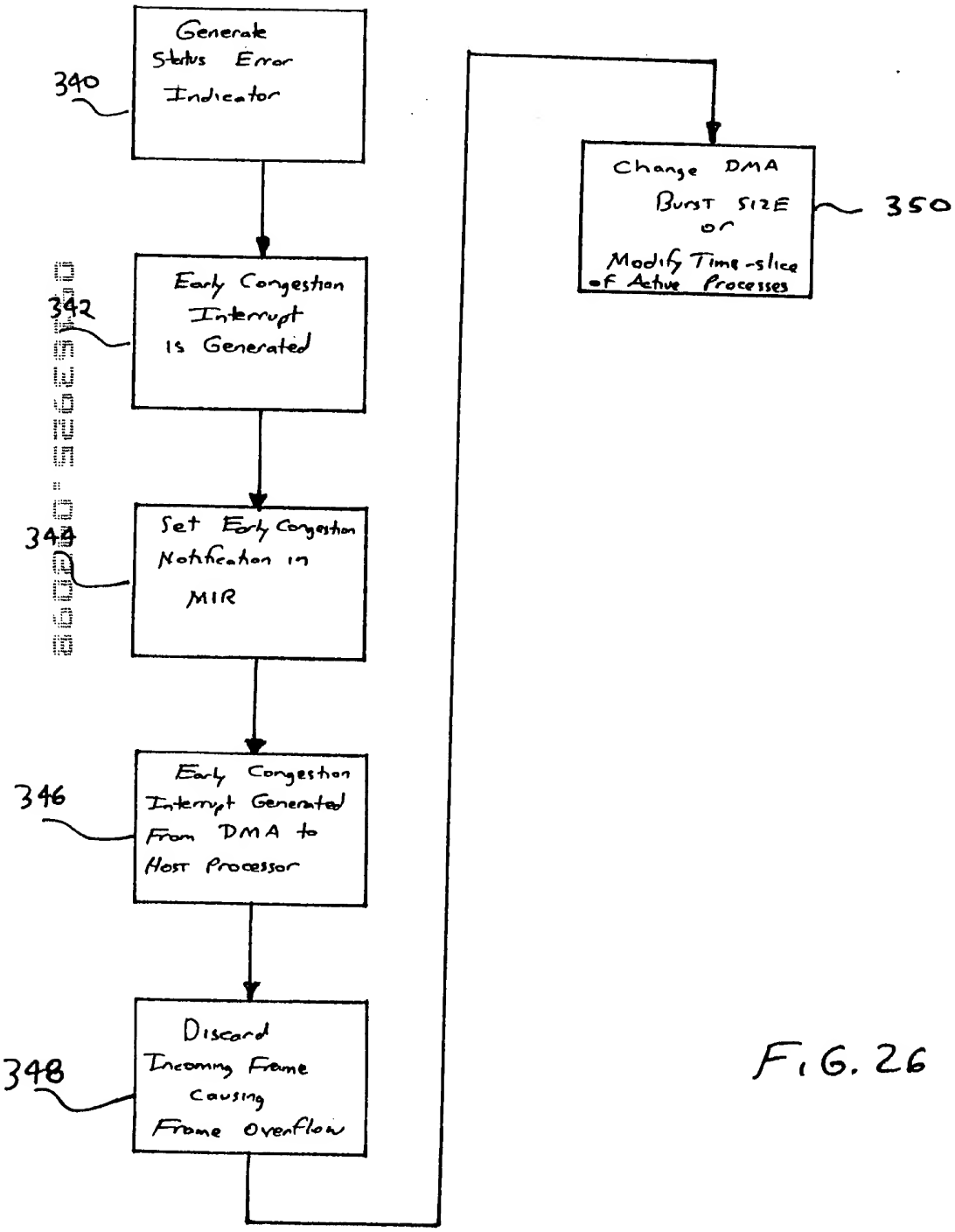

Generate Status Error Indicator
340
Early Congestion Interrupt Is Generated
342
Set Early Congestion Notification in MIR
344
Early Congestion Interrupt Generated From DMA to Host Processor
346
Discard Incoming Frame Causing Frame Overflow
348
Change DMA Burst Size or Modify Time-slice of Active Processes
350

FIG. 26

FIFO Cases Overflow on Second Packet into Receive Fifo

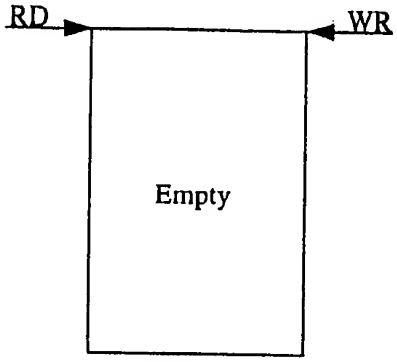


FIG. 27a

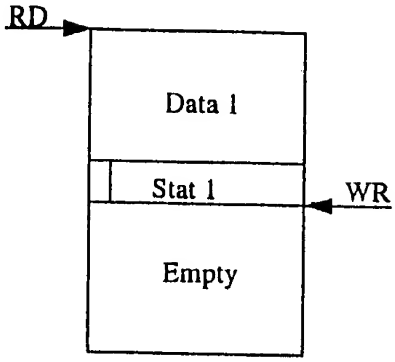


FIG. 27b

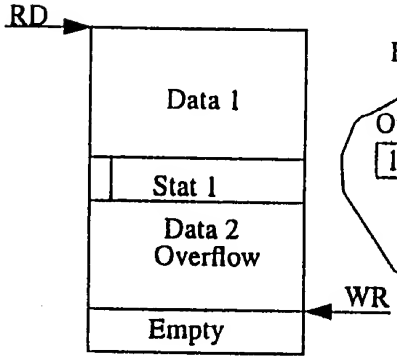


Fig 27c

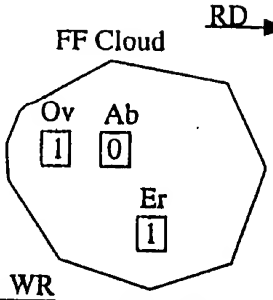


Fig. 27g

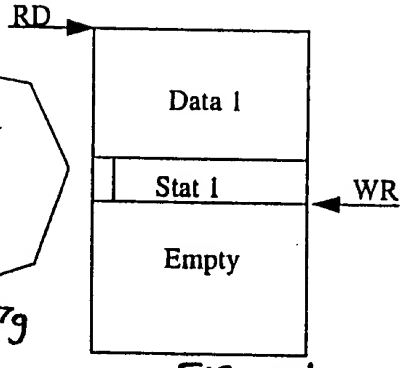


FIG. 27d

Write Pointer Reset to Beginning of Packet 2 Frozen until EOP Occurs at Which Time Error Status for Overflow Packet is Entered.

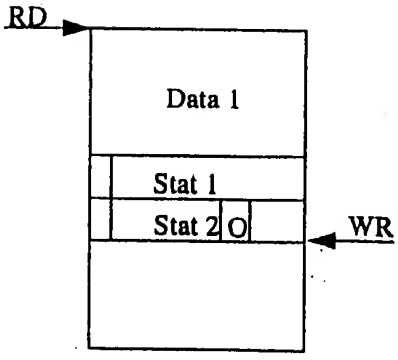
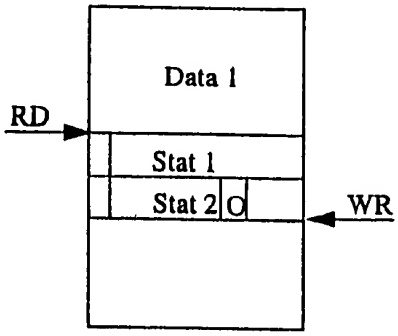


Fig. 27e



Read of Stat 1 by DMA copies it into Receive Status Register at Host Address. No requests to DMA for another data transfer will occur until CPC reads status. This prevents overwriting of status register by overflow status.

FIG. 27f

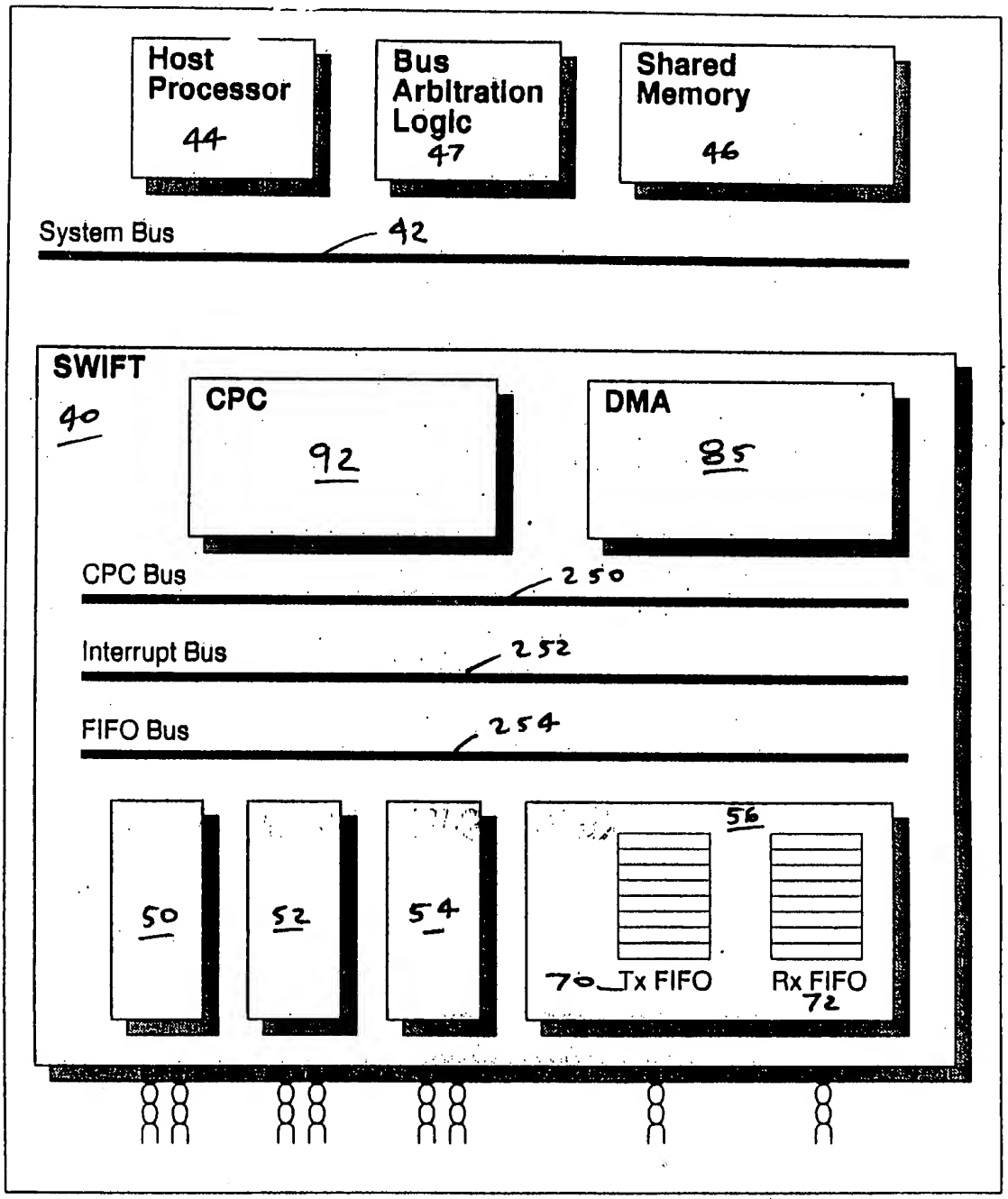

Host Processor
44
Bus Arbitration Logic
47
Shared Memory
46
System Bus
42
SWIFT
40
CPC
92
DMA
85
CPC Bus
250
Interrupt Bus
252
FIFO Bus
254
50
52
54
56
70
Tx FIFO
Rx FIFO
72

FIG. 28

Rx FIFO Begins to Fill with a New Packet

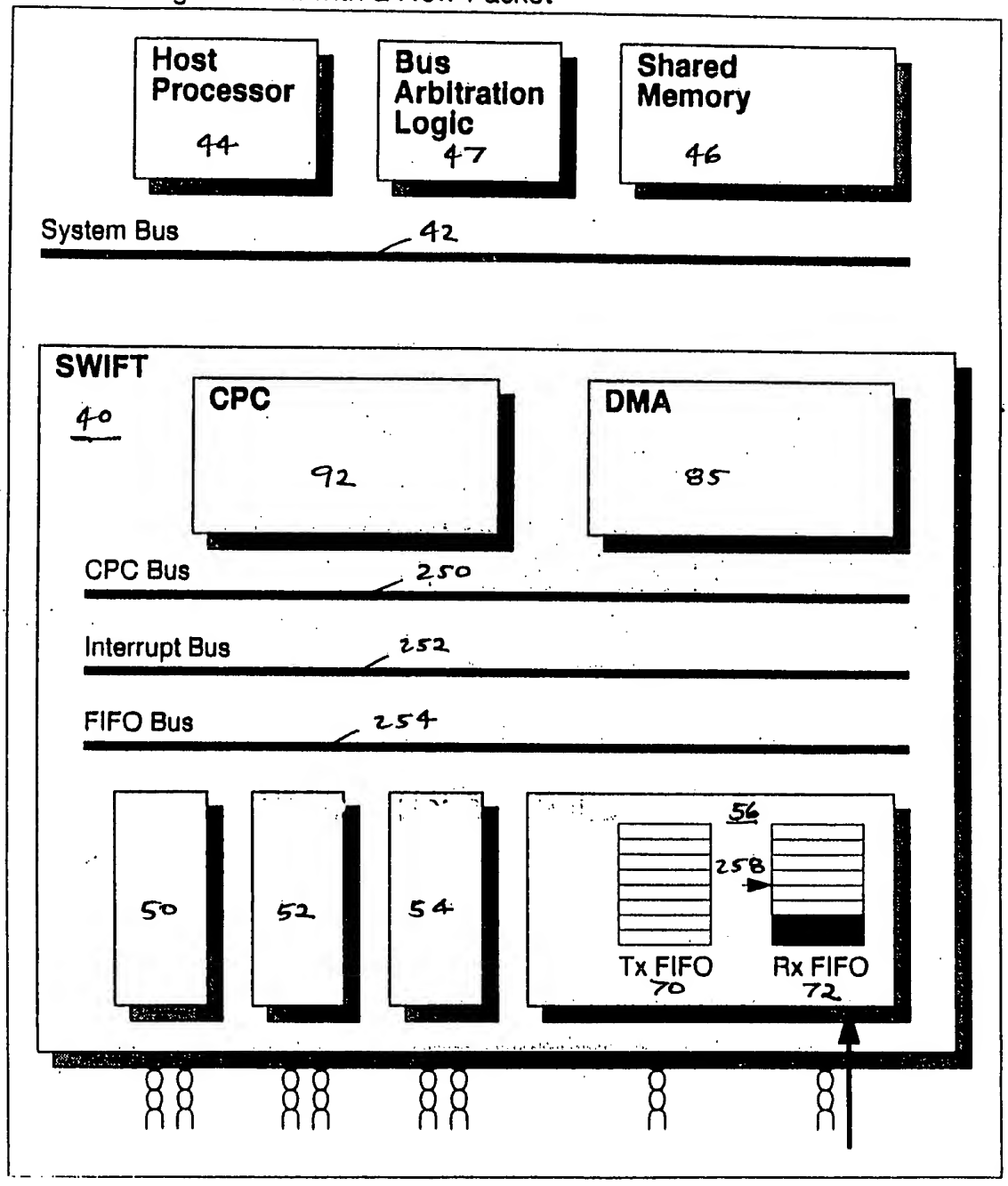


FIG. 29

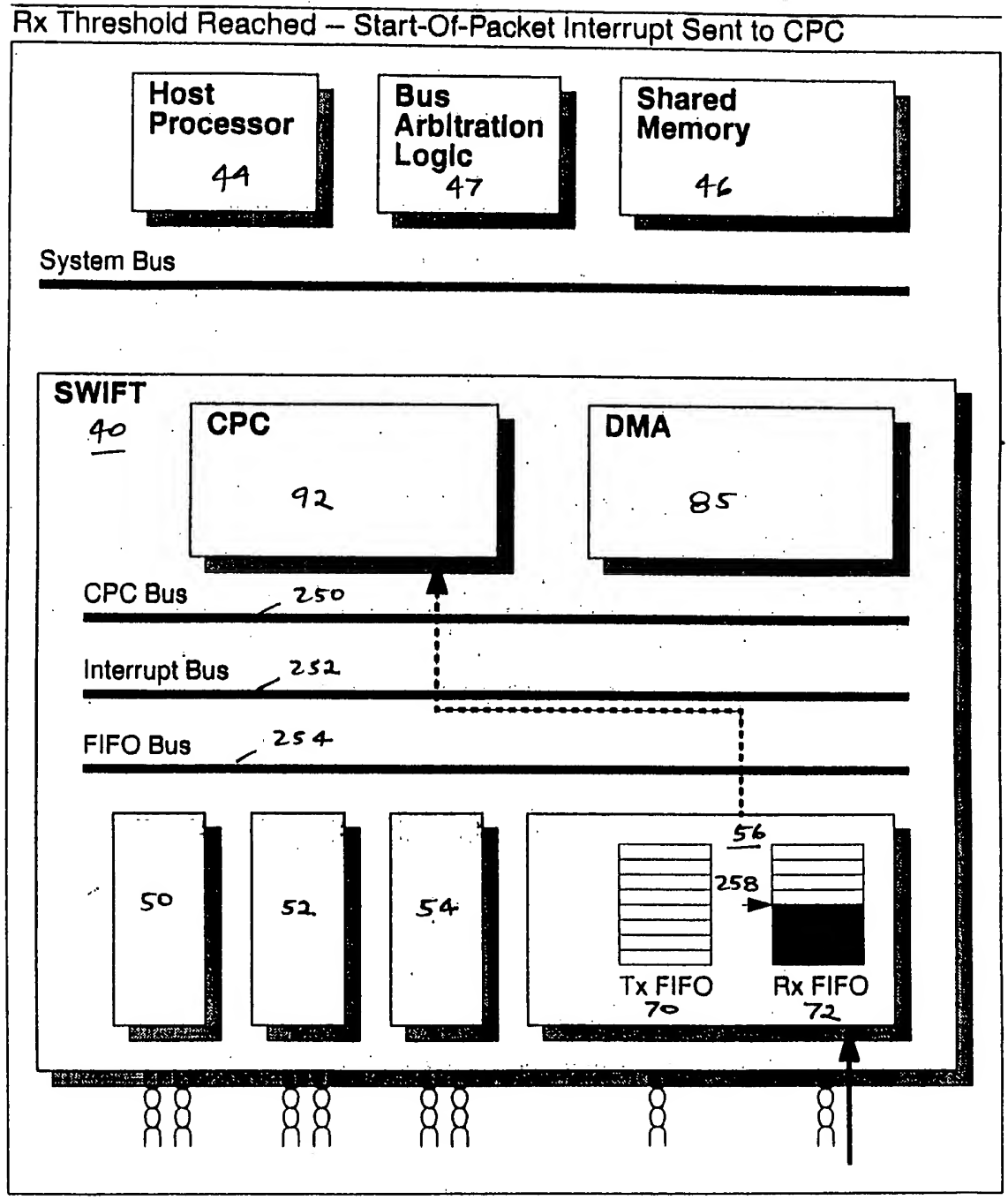

Rx Threshold Reached – Start-Of-Packet Interrupt Sent to CPC
Host Processor
44
Bus Arbitration Logic
47
Shared Memory
46
System Bus
SWIFT
40
CPC
92
DMA
85
CPC Bus
250
Interrupt Bus
252
FIFO Bus
254
50
52
54
56
258
Tx FIFO
70
Rx FIFO
72

FIG. 3D

CPC Issues a Command to DMA to Transfer Data

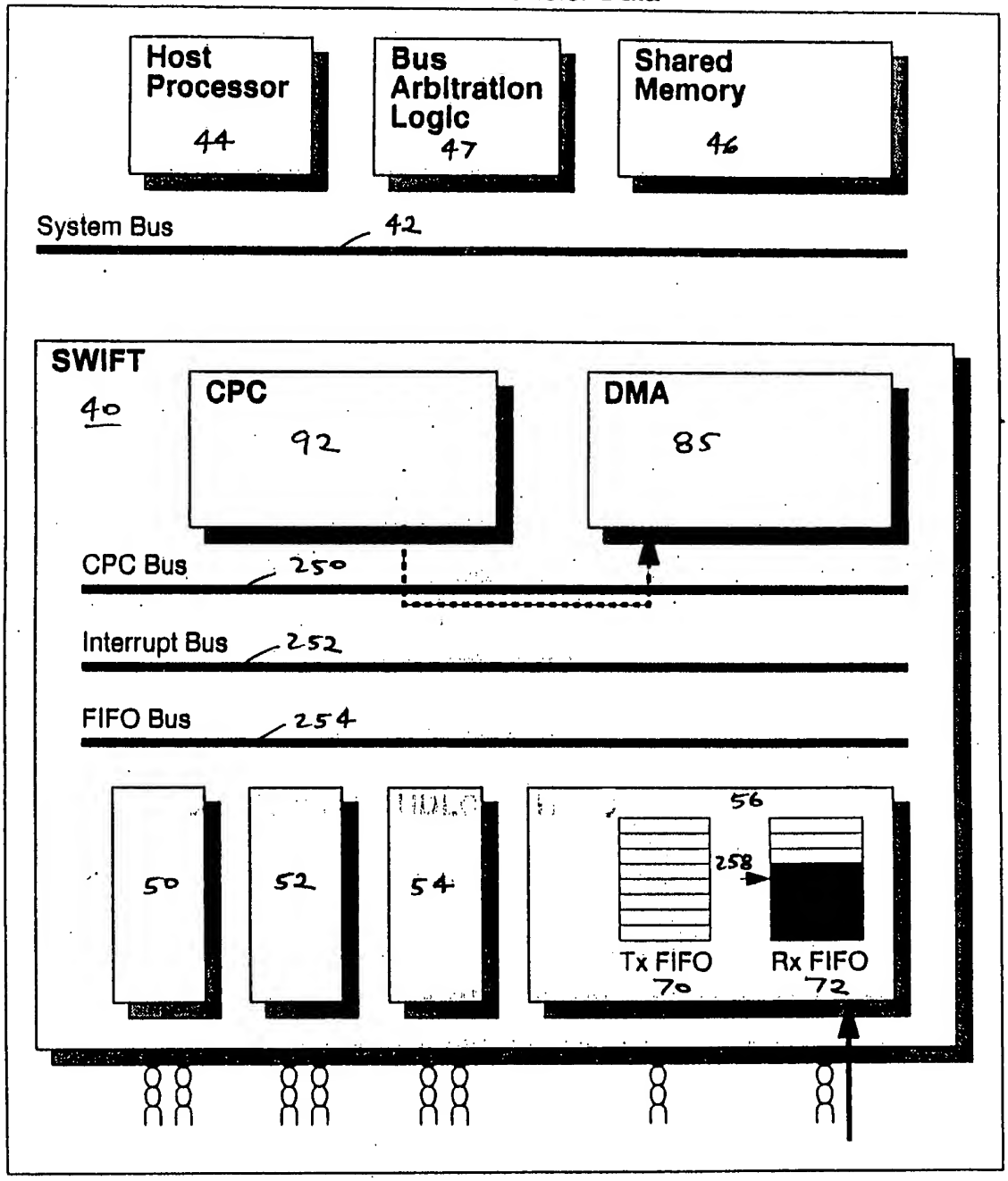


FIG. 31

DMA Negotiates for Ownership of System Bus

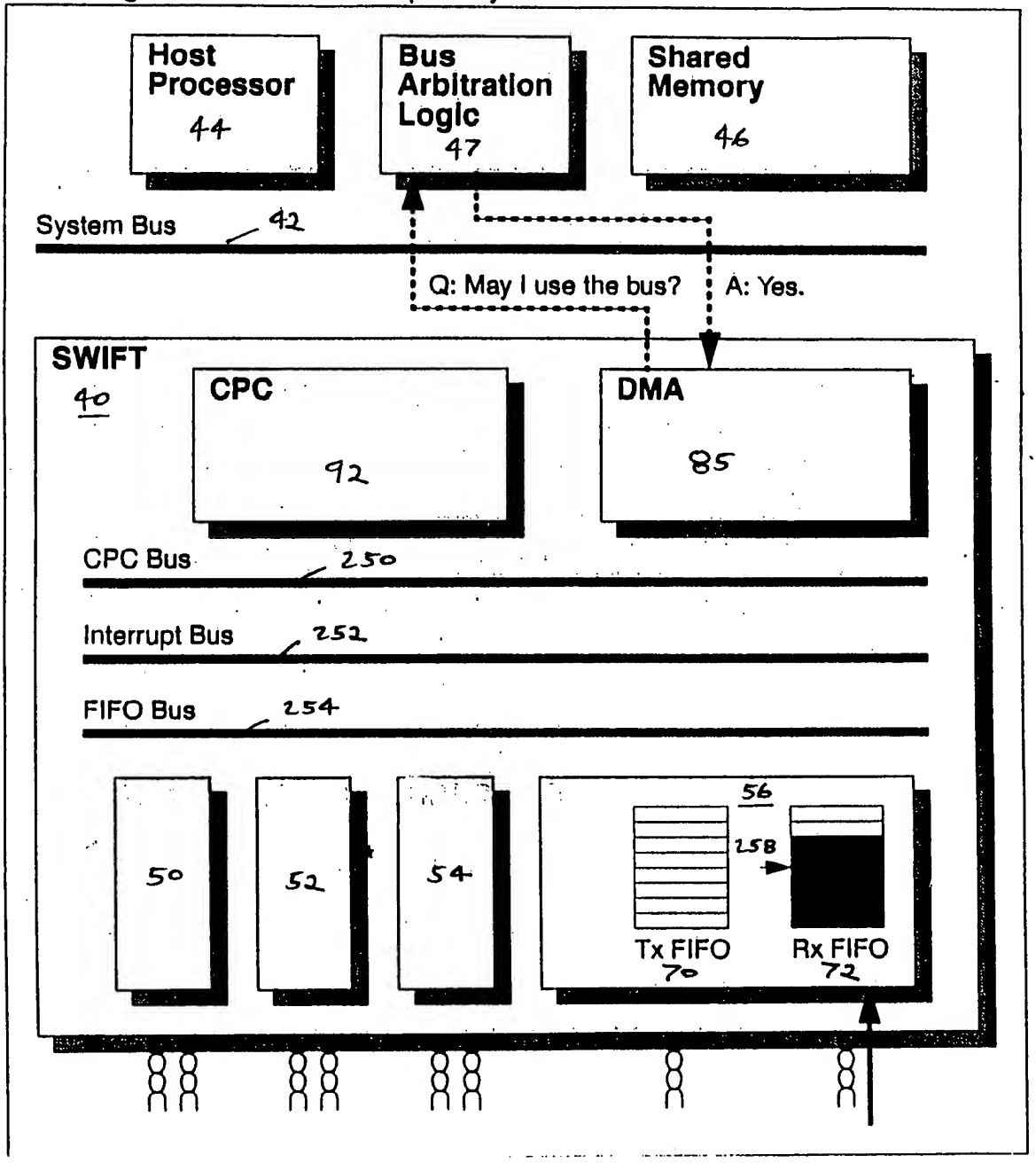


FIG. 32

DMA Transfers Data from Rx FIFO to Shared System Memory

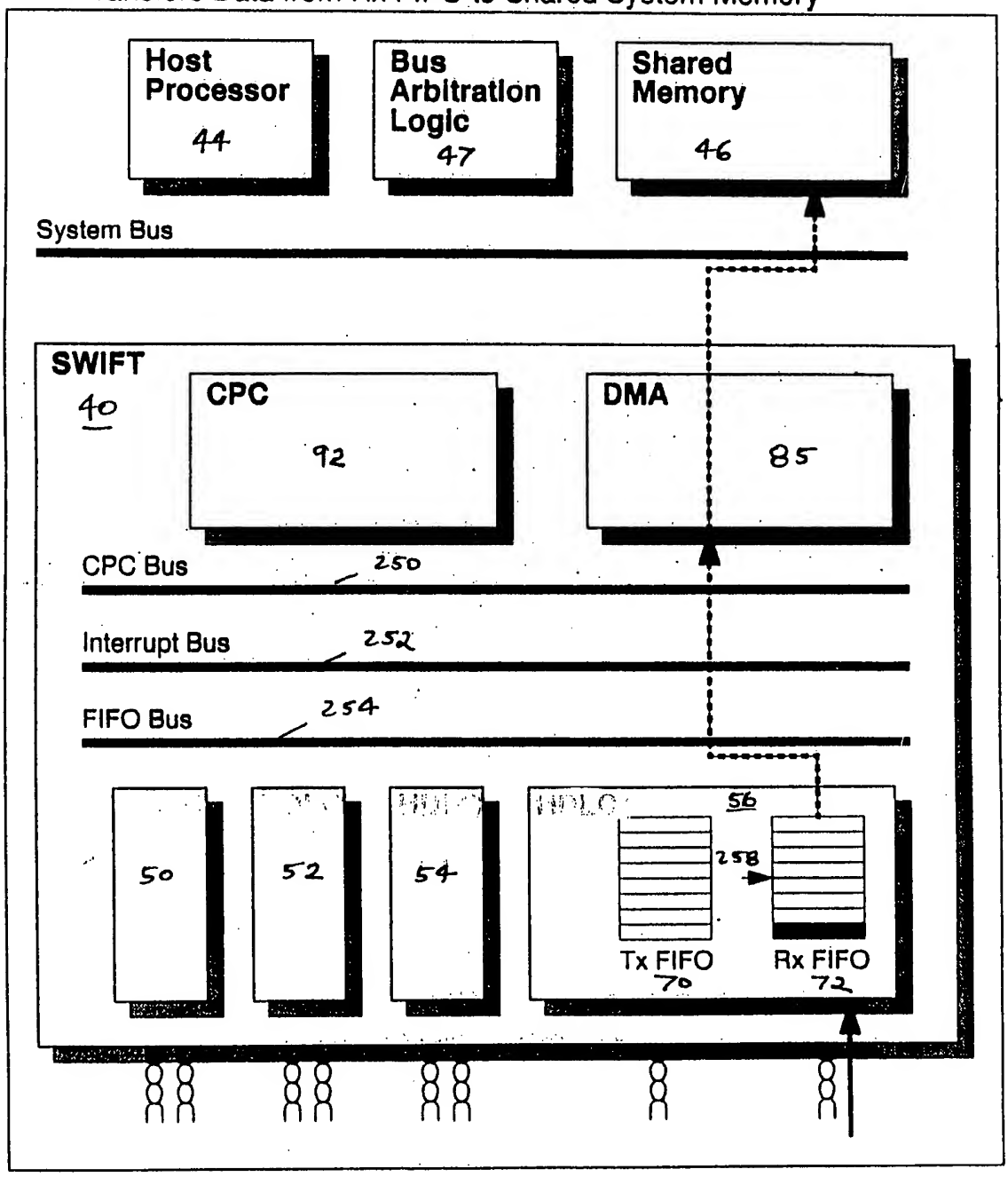


FIG. 33

Rx FIFO Begins to Fill with a New Packet 2

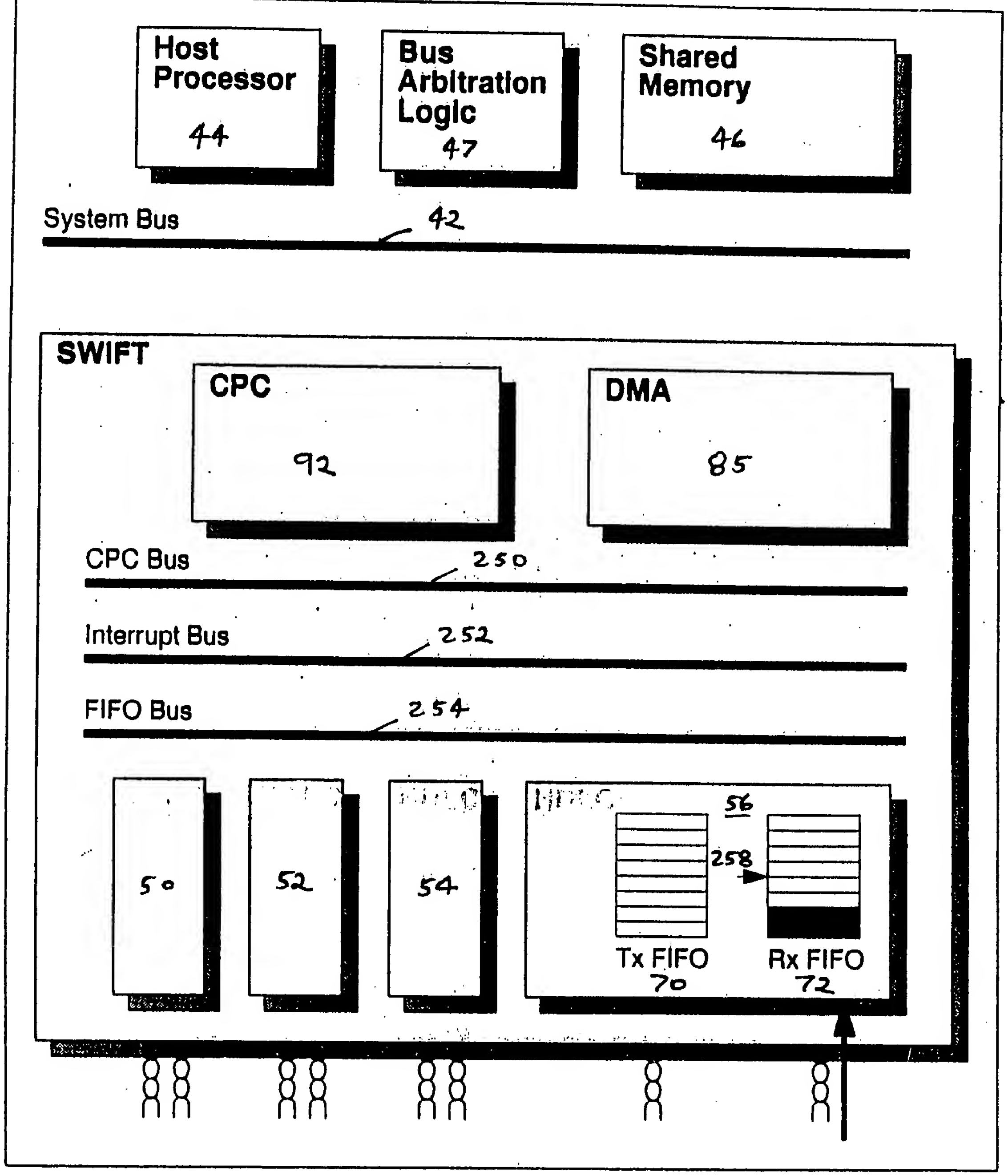


FIG. 34

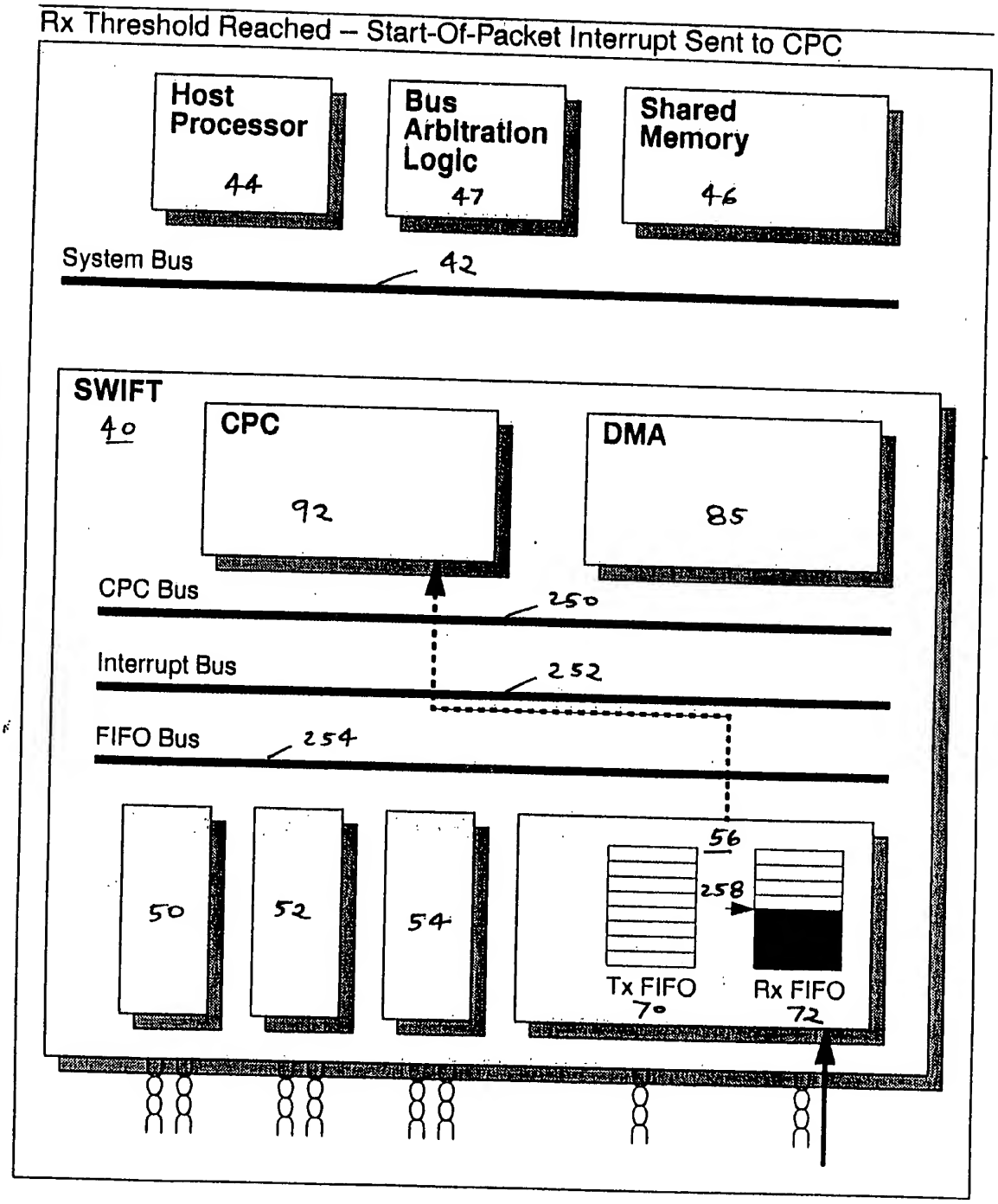
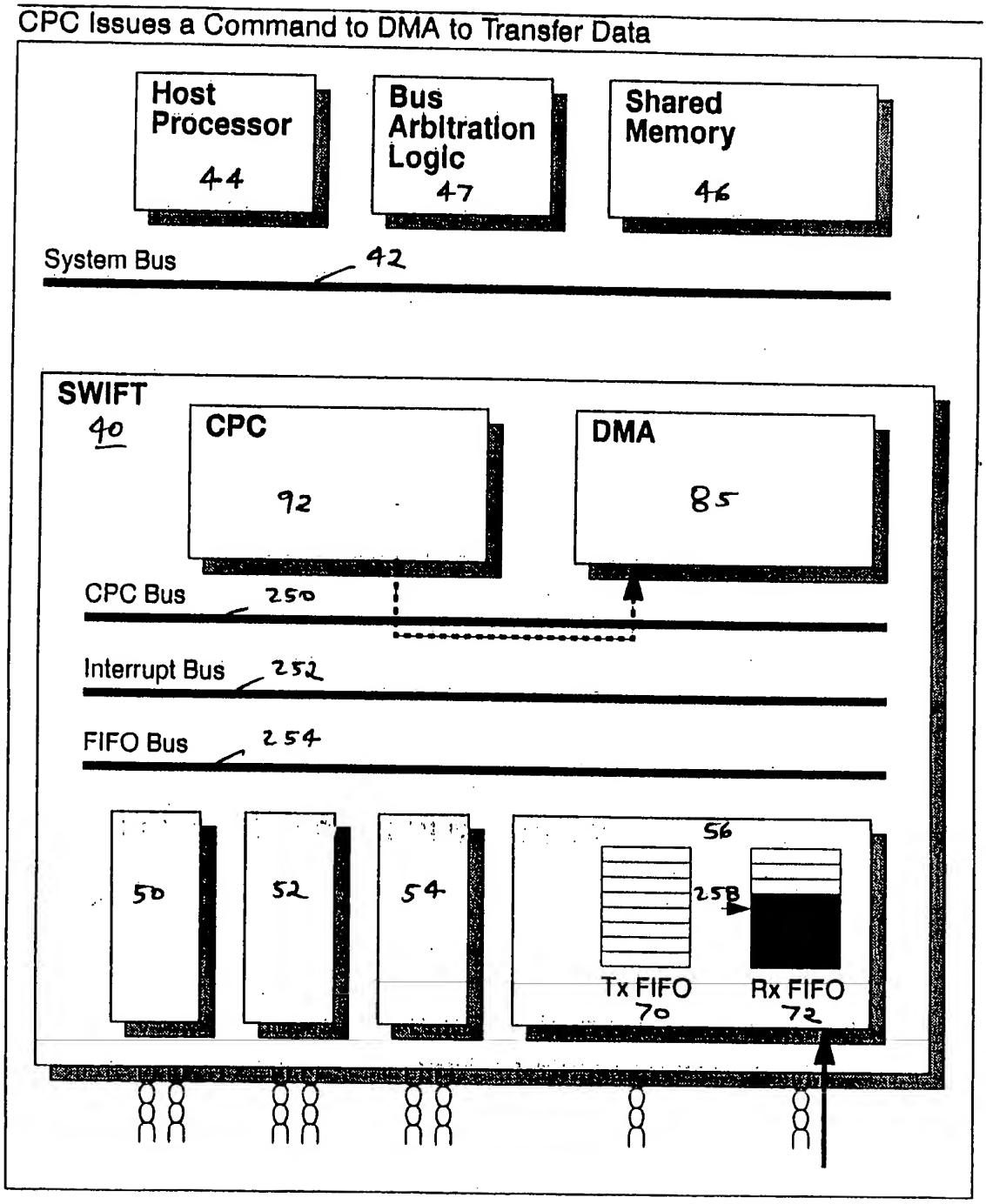
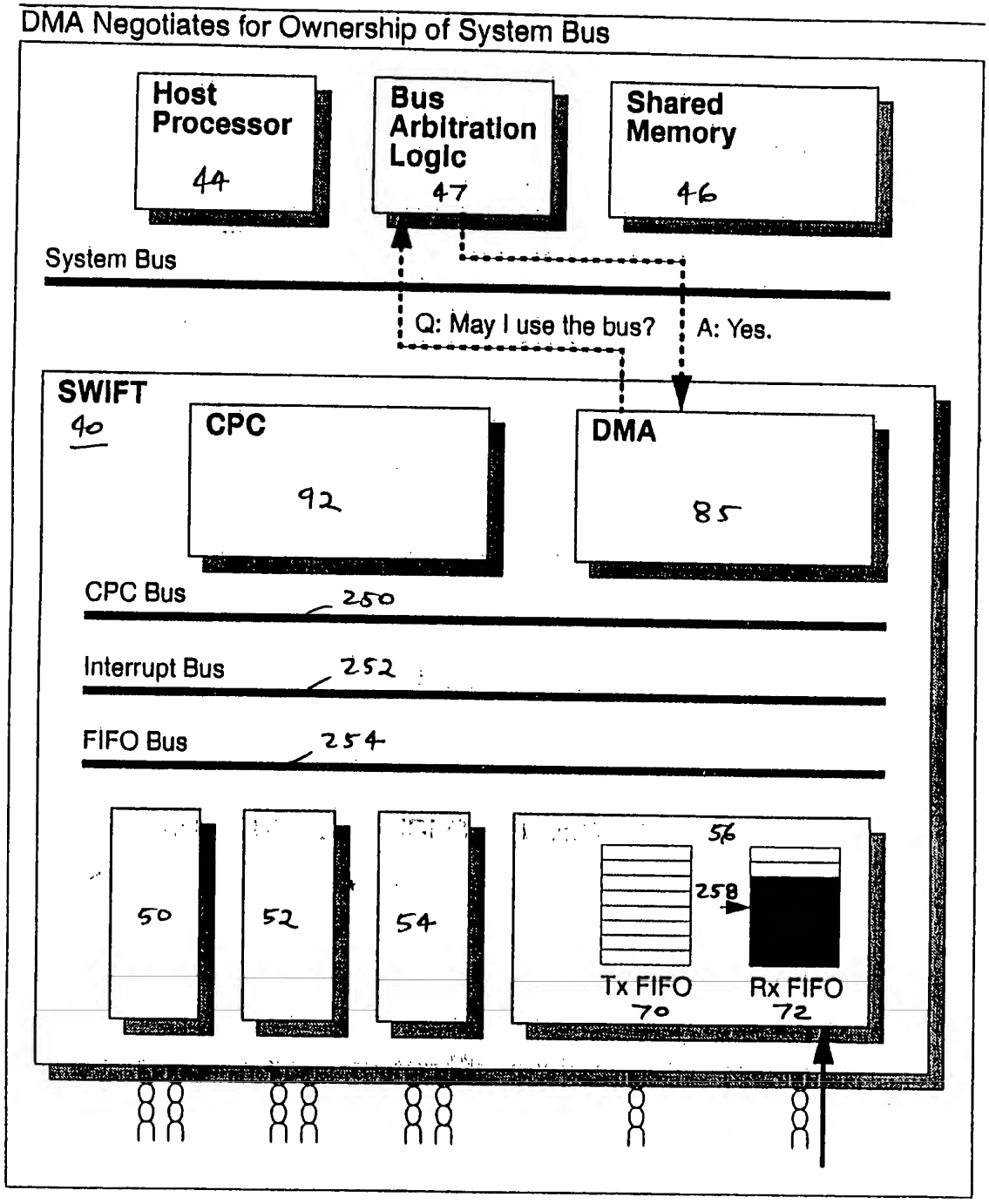

Rx Threshold Reached – Start-Of-Packet Interrupt Sent to CPC
Host Processor
44
Bus Arbitration Logic
47
Shared Memory
46
System Bus
42
SWIFT
40
CPC
92
DMA
85
CPC Bus
250
Interrupt Bus
252
FIFO Bus
254
50
52
54
56
258
Tx FIFO
70
Rx FIFO
72

FIG. 35


CPC Issues a Command to DMA to Transfer Data
Host Processor
44
Bus Arbitration Logic
47
Shared Memory
46
System Bus
42
SWIFT
40
CPC
92
DMA
85
CPC Bus
250
Interrupt Bus
252
FIFO Bus
254
50
52
54
56
25B
Tx FIFO
70
Rx FIFO
72

F.G. 36


DMA Negotiates for Ownership of System Bus
Host Processor
44
Bus Arbitration Logic
47
Shared Memory
46
System Bus
Q: May I use the bus?
A: Yes.
SWIFT
40
CPC
92
DMA
85
CPC Bus
250
Interrupt Bus
252
FIFO Bus
254
50
52
54
56
258
Tx FIFO
70
Rx FIFO
72

F.G. 37

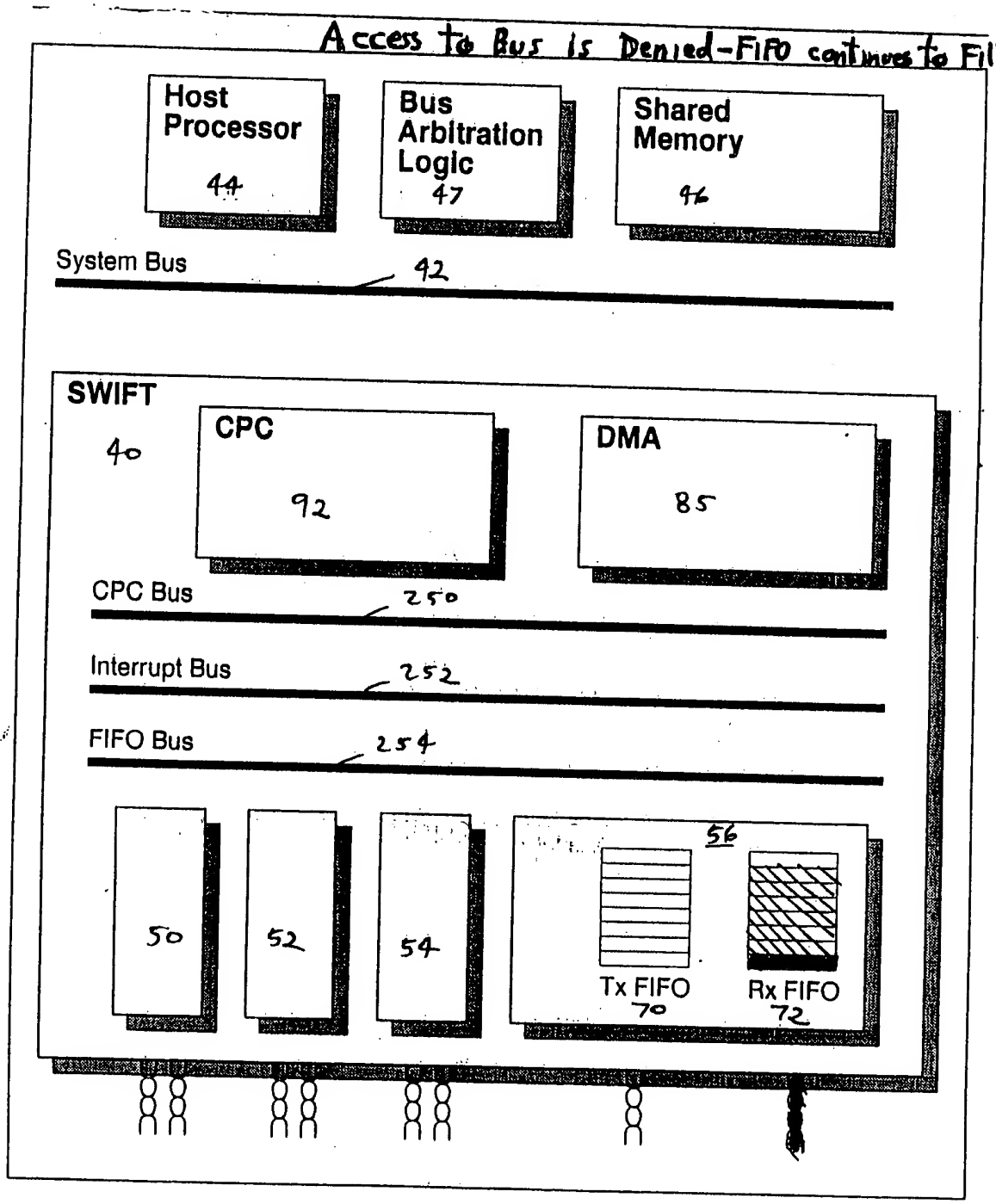

Access to Bus is Denied-FIFO continues to Fill #3
Host Processor
44
Bus Arbitration Logic
47
Shared Memory
46
System Bus
42
SWIFT
40
CPC
92
DMA
85
CPC Bus
250
Interrupt Bus
252
FIFO Bus
254
50
52
54
56
Tx FIFO
70
Rx FIFO
72

FIG. 38

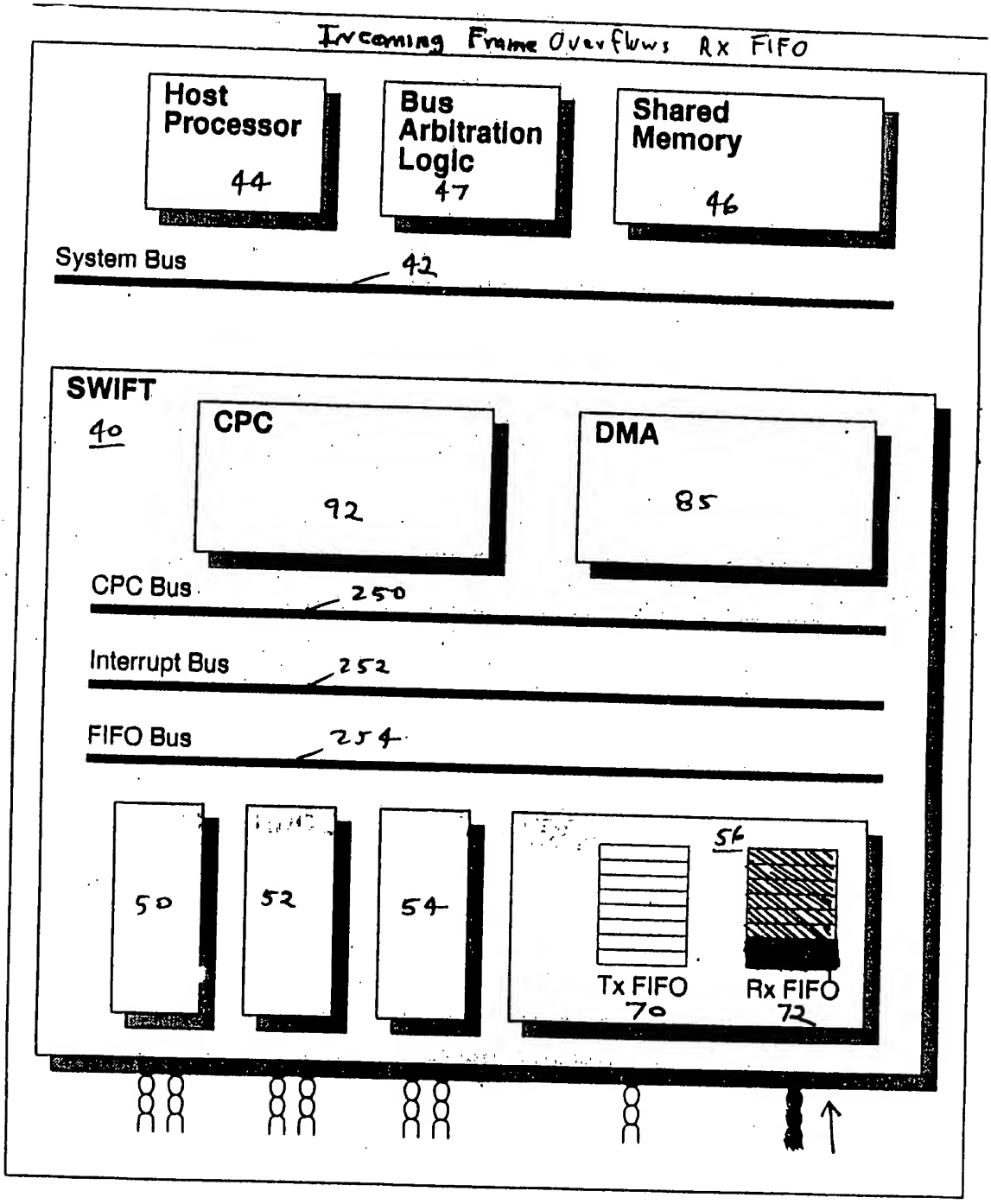

Incoming Frame Overflows Rx FIFO
Host Processor
44
Bus Arbitration Logic
47
Shared Memory
46
System Bus
42
SWIFT
40
CPC
92
DMA
85
CPC Bus
250
Interrupt Bus
252
FIFO Bus
254
50
52
54
56
Tx FIFO
70
Rx FIFO
72

FIG. 39

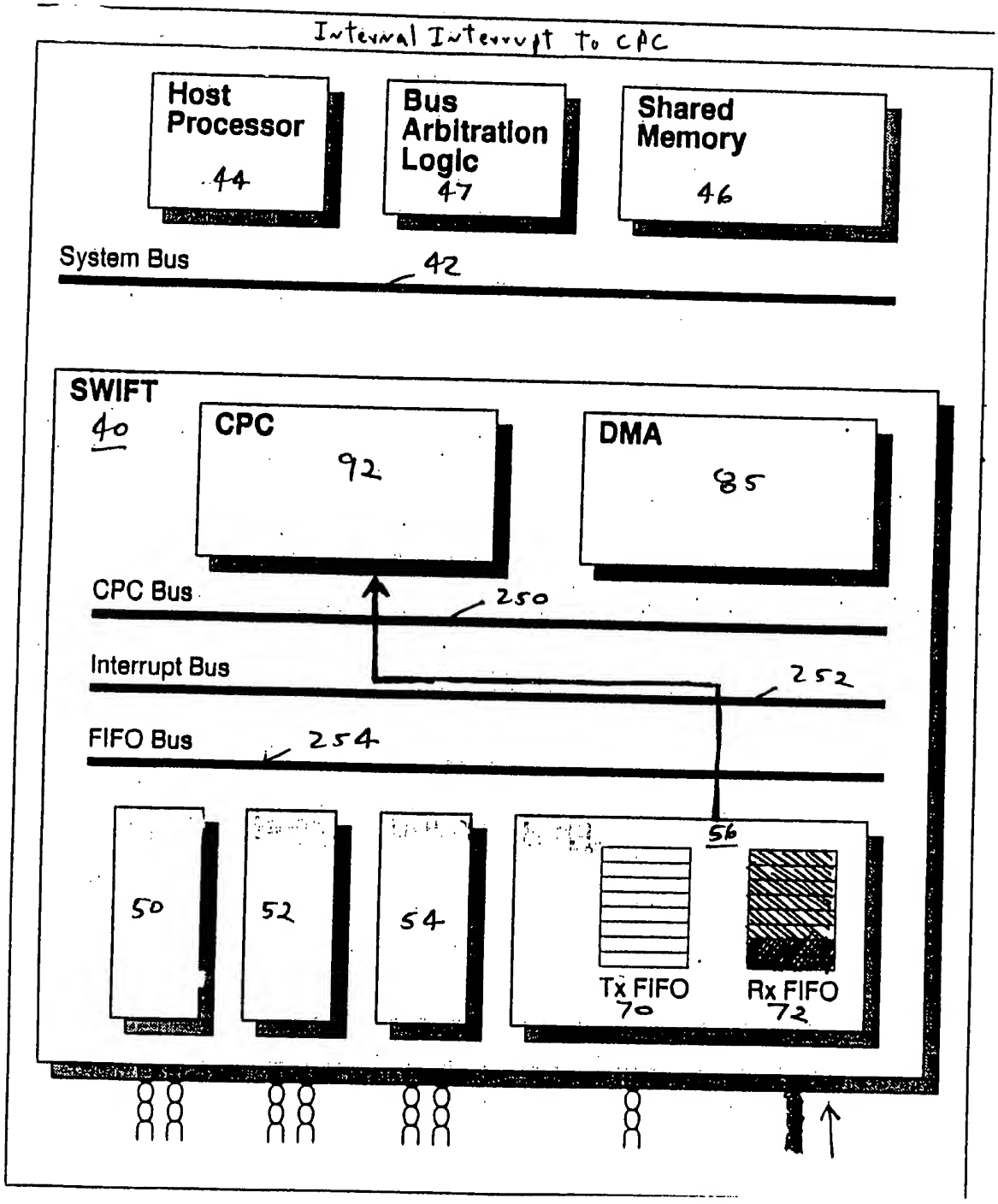

Internal Interrupt to CPC
Host Processor
44
Bus Arbitration Logic
47
Shared Memory
46
System Bus
42
SWIFT
40
CPC
92
DMA
85
CPC Bus
250
Interrupt Bus
252
FIFO Bus
254
50
52
54
56
Tx FIFO
70
Rx FIFO
72

FIG. 40

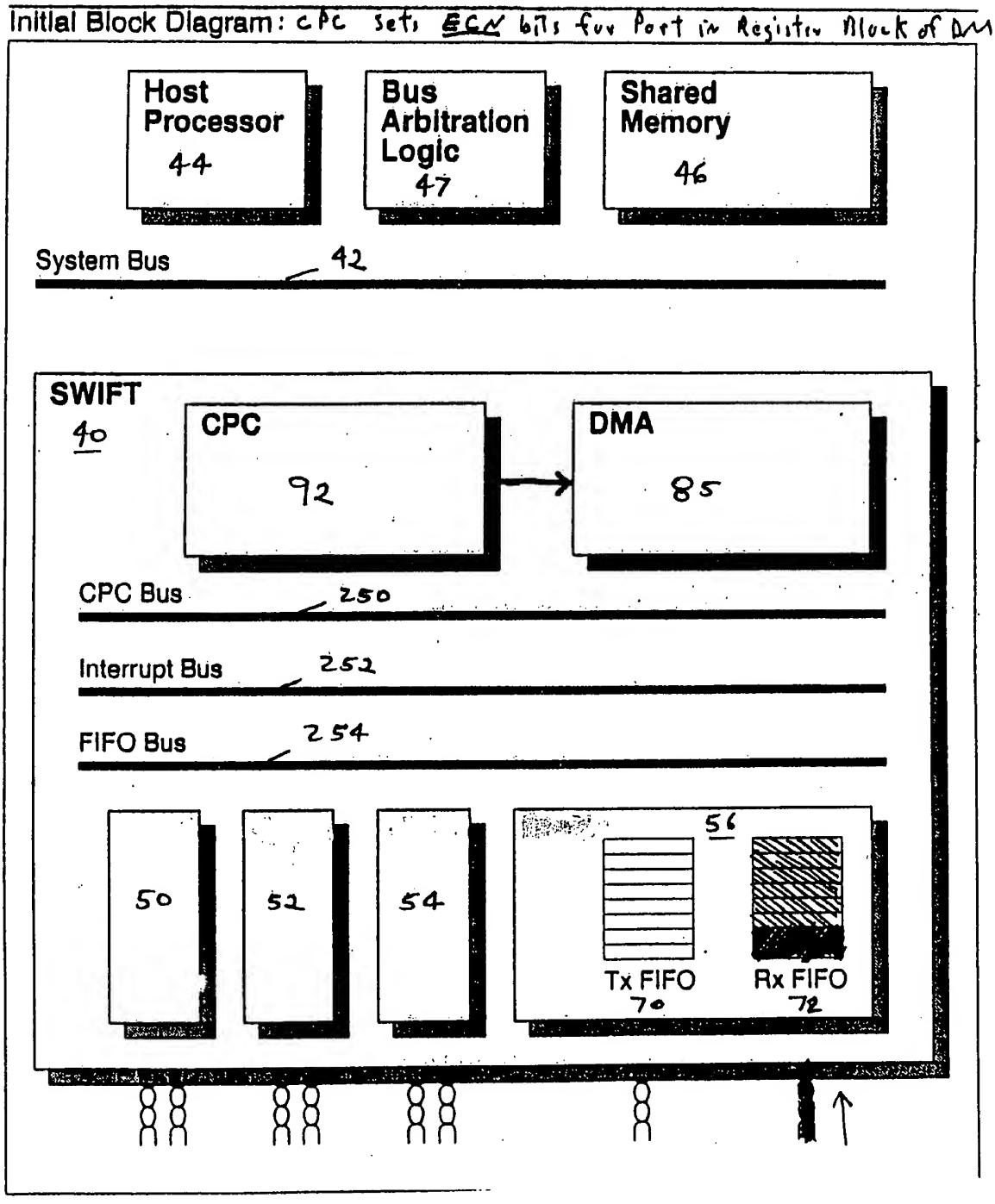
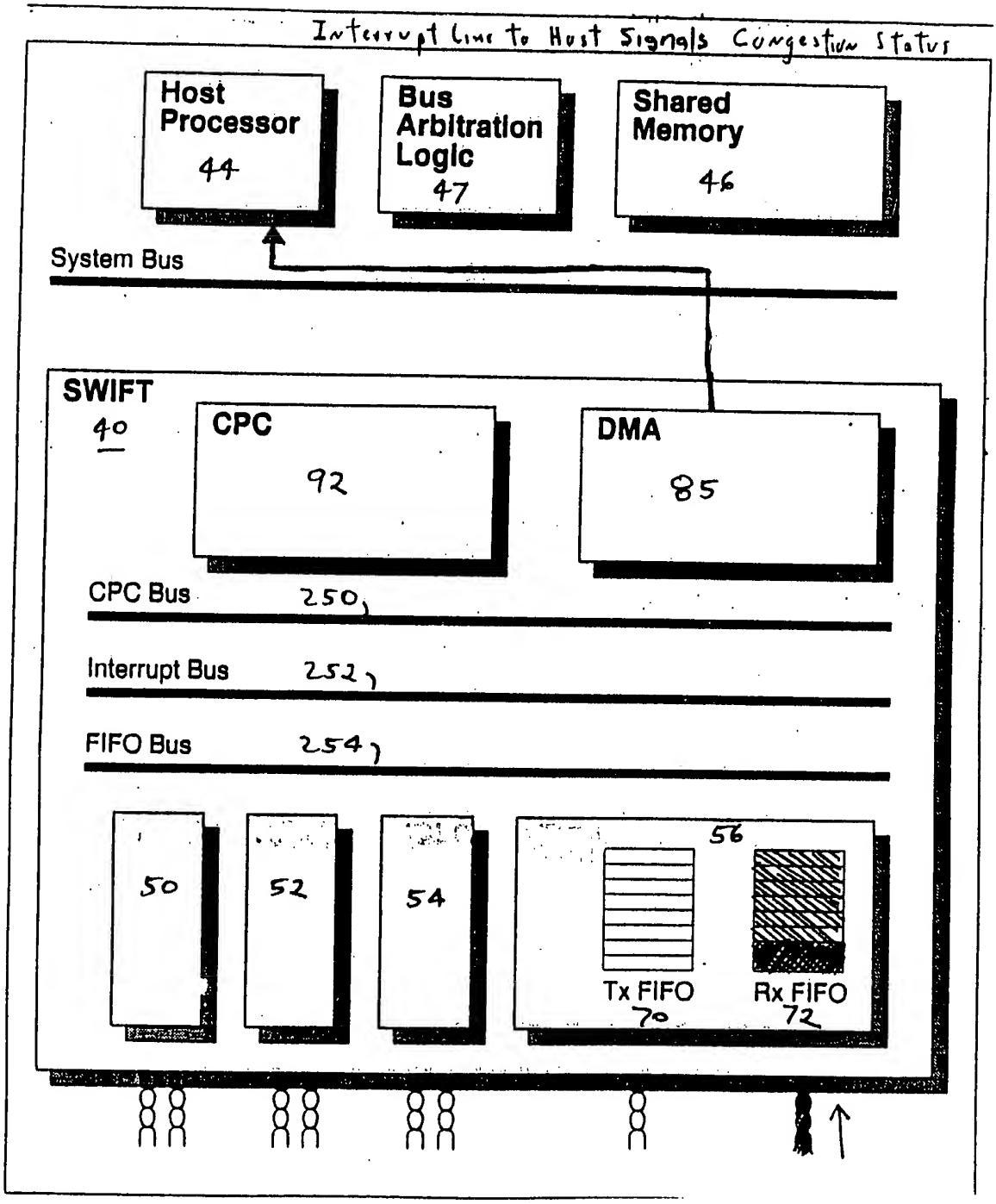

Initial Block Diagram: CPC sets ECN bits for Port in Register Block of DMA
Host Processor 44
Bus Arbitration Logic 47
Shared Memory 46
System Bus 42
SWIFT 40
CPC 92
DMA 85
CPC Bus 250
Interrupt Bus 252
FIFO Bus 254
50
52
54
56
Tx FIFO 70
Rx FIFO 72

FIG. 41


Interrupt Line to Host Signals Congestion Status
Host Processor 44
Bus Arbitration Logic 47
Shared Memory 46
System Bus
SWIFT 40
CPC 92
DMA 85
CPC Bus 250
Interrupt Bus 252
FIFO Bus 254
50
52
54
56
Tx FIFO 70
Rx FIFO 72

F.G.42

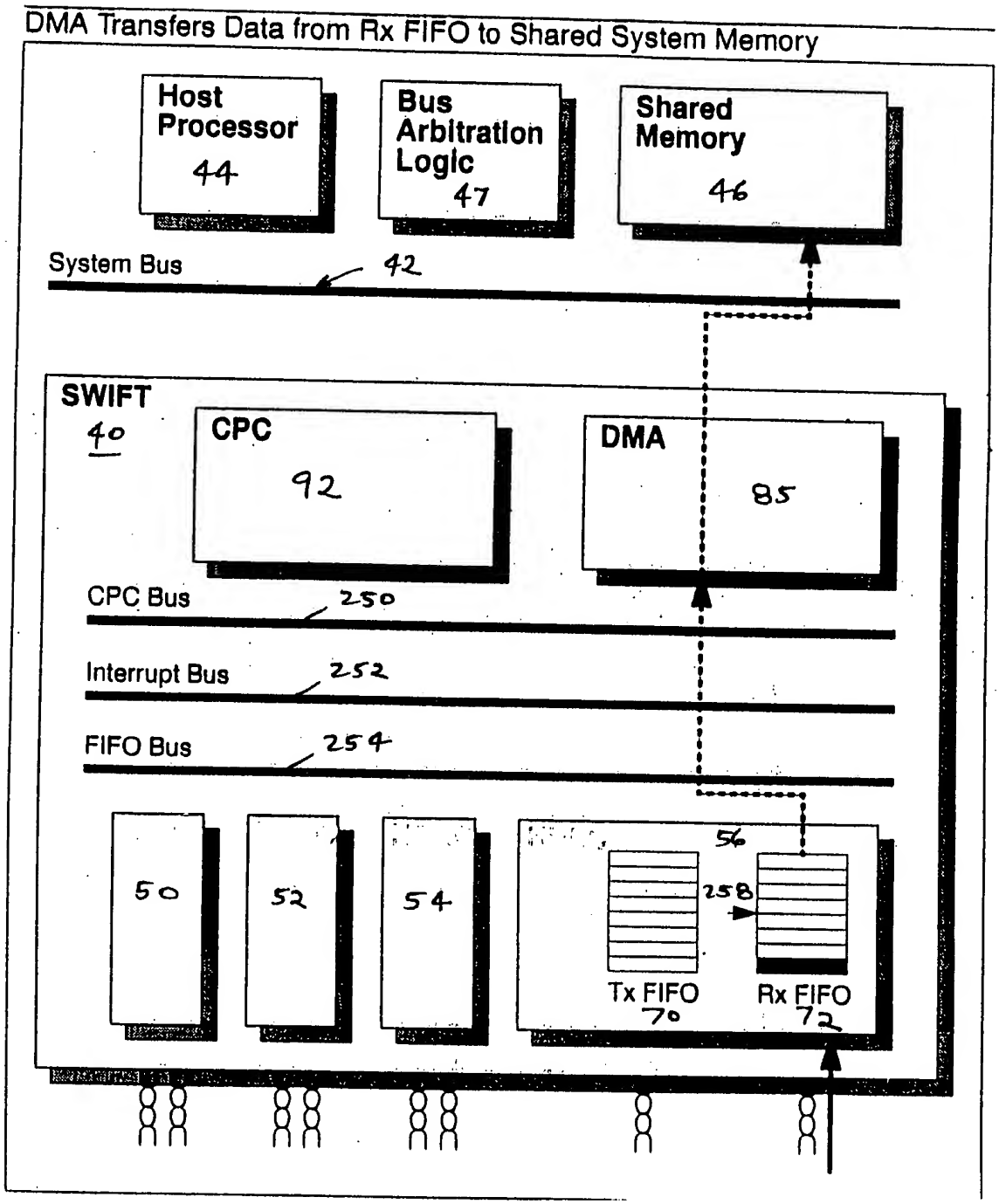

DMA Transfers Data from Rx FIFO to Shared System Memory
Host Processor
44
Bus Arbitration Logic
47
Shared Memory
46
System Bus
42
SWIFT
40
CPC
92
DMA
85
CPC Bus
250
Interrupt Bus
252
FIFO Bus
254
50
52
54
56
258
Tx FIFO
70
Rx FIFO
72

FIG. 43

Estimated Traffic Composition of the Host Bus

DATA: X/32
DESCRIPTORS: 2*X/128 + 2
TOTAL: X/32 + 2*X/128 + 2

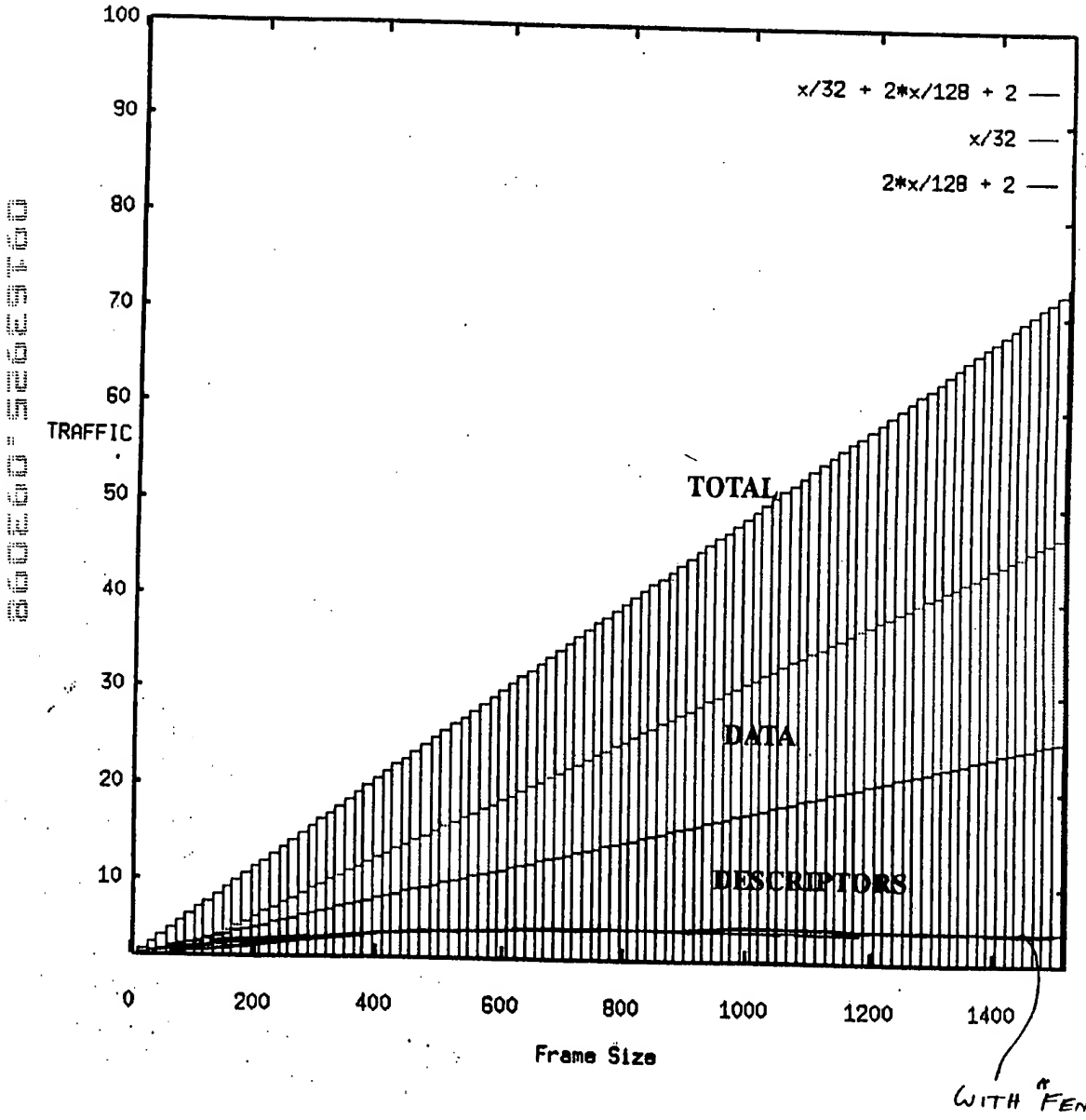


FIG. 44

Primitive Signalling

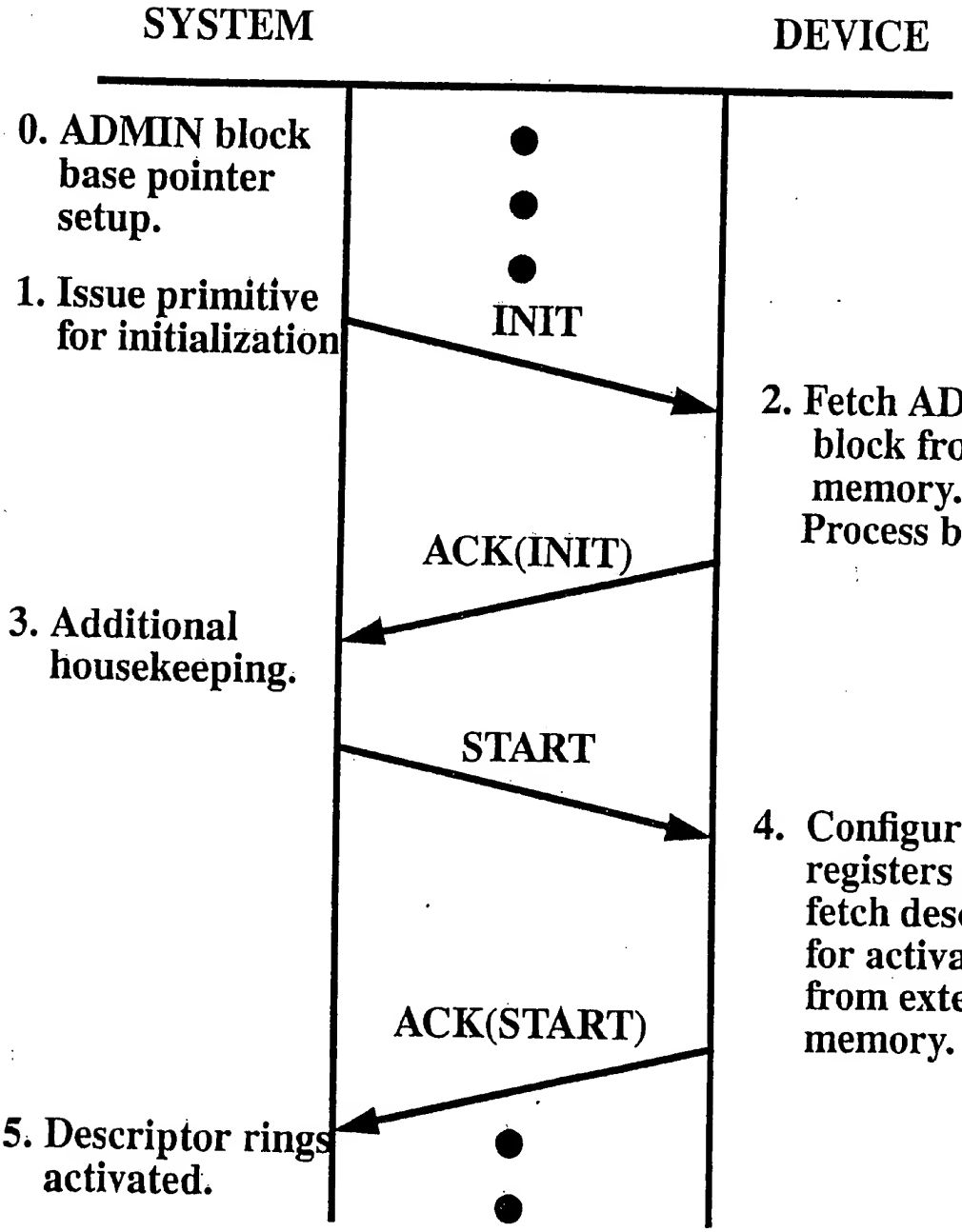


FIG. 45

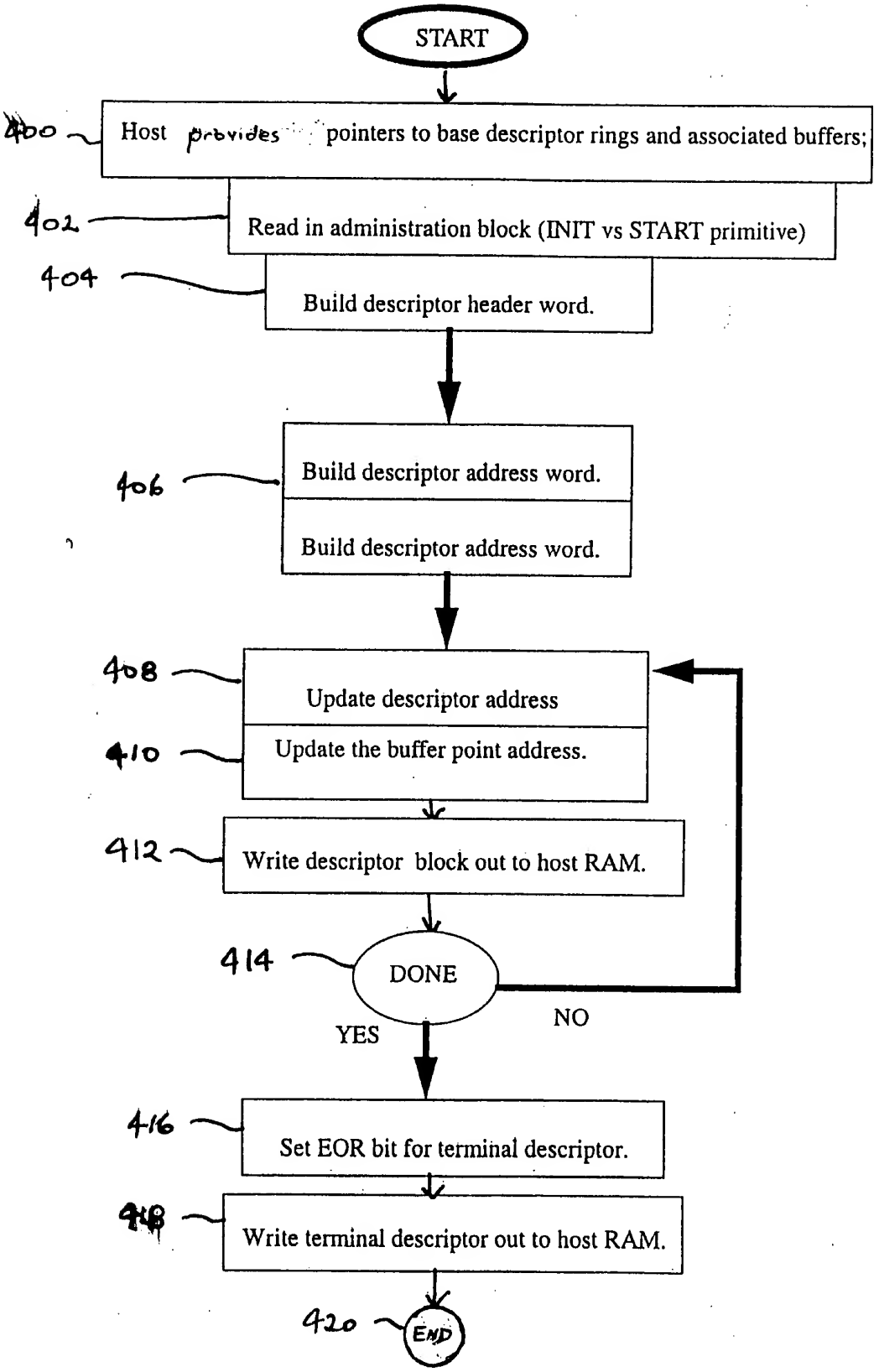

START
400
Host provides pointers to base descriptor rings and associated buffers;
402
Read in administration block (INIT vs START primitive)
404
Build descriptor header word.
406
Build descriptor address word.
Build descriptor address word.
408
Update descriptor address
410
Update the buffer point address.
412
Write descriptor block out to host RAM.
414
DONE
NO
YES
416
Set EOR bit for terminal descriptor.
418
Write terminal descriptor out to host RAM.
420
END

FIG.46

Receive Message Descriptor 0

RMD0	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Name	OWN	EOR	ENP	BSIZE[12:2]											RERR	ROFLO	MSIZE[12:0]															

Bit #	Field	Name	Description
31	OWN	Descriptor Ownership	**(1=Device; 0=Host)** Establishes ownership of the Receive Message Descriptor and its associated data buffer. The OWN bit is used as a handshake between and the host. No part of the Receive Message Descriptor or the contents of its associated buffer should be altered once ownership has been relinquished.
30	EOR	End of Ring	**(1=end-of-ring; 0=not end-of-ring)** Denotes the last Receive Message Descriptor in the descriptor ring. Causes Device to return to the top of the ring after using this descriptor. In other words, the next descriptor used by Device will be the first entry in the ring.
29	ENP	End of frame	**(1=end-of-frame; 0=chain)** Indicates the associated receive data buffer contains the end of a received frame. ENP of zero implies buffer 'chaining' where the received frame spans two or more, adjacent descriptors.
28:18	BSIZE	Buffer Size	**(10-bit unsigned integer)** Indicates the number of bytes available in the associated receive data buffer (up to 8K bytes). Note that buffers are allocated in 4-byte (1-word) increments since the BSIZE field is defined as bits 12 to 2. The BSIZE field is positioned in the upper half-word to facilitate this definition. A BSIZE of zero defaults to a buffer size of one word. The actual number of bytes available in a buffer are determined by the BSIZE field and the starting address of the buffer (RBADR). Receive data buffers are permitted to start on any byte address but are always assumed by Device to end on a word-aligned boundary. In other words, the last address of every receive buffer is a complete, 4-byte word.
17	RERR	Rx Error Summary	**(1=error; 0=normal)** Logical OR summary of the error status bits reported in the receive status word written by Device into the first full word following the end of the frame in the buffer. RERR summarizes: CNTOVL, FCS, RABRT, ROFLO. Allows a single-bit test for receive frame-related errors.
16	ROFLO	Rx FIFO Overflow Error	**(1=error; 0=normal)** Indicates a dropped packet due to insufficient space available in the receive FIFO. When overflow occurs the HDLC unit continues to monitor the incoming packet for statistical purposes, and drops the entire packet (or at least the portion not yet read from the FIFO). The resulting status word is written into the FIFO along with the End-Of-Packet tag. Overflow is caused by inadequate servicing (reading) of the FIFO. If this bit is set MSIZE may not indicate the actual amount of data in the buffer.
15:0	MSIZE	Message Size	**(15-bit unsigned integer)** Indicates the number of octets occupied by part or all of a received frame in the associated buffer. MSIZE does not include the four octets of the receive status word written by Device into the first full word following the end of the frame in the buffer. The MSIZE field is expected to be all zeros when the host gives ownership of the descriptor to Device. Since no attempt is made by Device to check this, any non-zero value given will result in an erroneous MSIZE returned.

FIG. 47

Receive Message Descriptor 1

RMD1	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Name	RBADR[31:0]																															

Bit #	Field	Name	Description
31:0	RBADR	Receive Buffer Starting Address	**(32-bit unsigned integer)** Acts as a pointer to the first address location of the associated receive data buffer. Receive data buffers are used by Device to store incoming frames. No more than one frame is stored in a given buffer. A single frame may span multiple buffers when its size exceeds the buffer size. RBADR is a byte address in a 32-bit data word system implying that receive buffers are not required to begin on word-aligned boundaries. The rule imposed by Device is that receive buffers may start with any byte alignment, but always end on word-aligned boundaries. The following table outlines the byte alignment indicated by the least two significant RBADR bits. RBADR[1:0] — Valid Bytes — Alignment 00 — 4 — Aligned (full word) 01 — 3 — Non-aligned 10 — 2 — Non-aligned 11 — 1 — Non-aligned

FIG. 48

Transmit Message Descriptor 0

TMD0	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Name	OWN	EOR	ENP	NOCRC	TOFLO	*Reserved*										UFLO	MSIZE[12:0]															

Bit #	Field	Name	Description
31	OWN	Descriptor Ownership	**(1=Device; 0=Host)** Set by the host, cleared by Device. Establishes ownership of the Transmit Message Descriptor and its associated data buffer. The OWN bit is used as a handshake between Device and the host. No part of the Transmit Message Descriptor or the contents of its associated buffer should be altered once ownership has been relinquished.
30	EOR	End of Ring	**(1=end-of-ring; 0=not end-of-ring)** Configured by the host to mark the descriptor as the last entry in the ring. Denotes the last Transmit Message Descriptor in the descriptor ring. Causes Device to return to the top of the ring after using this descriptor. In other words, the next descriptor used by Device will be the first entry in the ring.
29	ENP	End of frame	**(1=end-of-frame; 0=not end-of-frame)** Set by the host to indicate that the associated transmit data buffer contains the end of a transmit frame. ENP of zero implies buffer 'chaining' where the frame to be transmitted spans two or more, adjacent descriptors.
28[a]	NOCRC	No CRC Appended	**(1=not appended; 0=appended)** Configured by the host to control Tx CRC generation on a per-frame basis. Prevents Frame Check Sequence (CRC) from being generated and appended automatically by the unit. NOCRC is only used by Device when the End of frame (ENP) bit is set.
27	TOFLO	TX FIFO Overflow error	**(1=error; 0=normal)** Set by HDLC when Fifo TX is in overflow. Probably due to watermark < burst size. This means an attempt has been made to write more than the available space in the FIFO TX.The only way to exit from this condition is to set TXFLUSH or Reset.
26:17	*Reserved*	–	Must be zero.
16	UFLO	Tx FIFO Underflow Error	**(1=error; 0=normal)** Set by Device when the transmit FIFO is emptied during a transmission before encountering the End-Of-frame. Underflow is caused by inadequate servicing (writing) of the FIFO.
15:0	MSIZE	Message Size	**(13-bit unsigned integer)** Set by the host to indicate the number of octets of a transmit frame contained in the associated transmit data buffer.

FIG. 49

Transmit Message Descriptor 1

TMD1	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Name	TBADR[31:0]																															

Bit #	Field	Name	Description
31:0	TBADR	Transmit Buffer Starting Address	**(32-bit unsigned integer)** Acts as a pointer to the first address location of the associated transmit data buffer. Transmit data buffers are used by Device as the source for outgoing frames. No more than one frame should be stored in a given buffer. A single frame may span multiple buffers when its size exceeds the buffer size. TBADR is a byte address in a 32-bit data word system implying that transmit buffers are not required to begin on word-aligned boundaries. The following table outlines the byte alignment indicated by the least two significant TBADR bits. TBADR[1:0] — Valid Bytes — Alignment 00 — 4 — Aligned (full word) 01 — 3 — Non-aligned 10 — 2 — Non-aligned 11 — 1 — Non-aligned

FIG. 50